THE UNIVERSITY OF ALBERTA

OPTIMAL PROCESS OPERATION IN THE FACE OF UNCERTAINTY

BY

© LINDA C. BISWANGER

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF CHEMICAL AND PETROLEUM ENGINEERING

EDMONTON, ALBERTA

FALL, 1972

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled "Optimal Process Operation in the Face of Uncertainty" submitted by Linda C. Biswanger in partial fulfilment of the requirements for the degree of Master of Science.

Date October 3, 1972

Abstract

An approach to the problem of optimal operation of an existing processing system in the face of uncertainty was suggested. An intrinsic concept is the restriction of the study to overall process relationships and operating conditions, leaving the optimization of individual unit operation for later, more detailed, studies as the need arises.

A simple, mathematically tractable, model of overall relationships is developed, enabling deterministic optimization with a minimum of effort. Critical features of the model are identified, and the extent of uncertainty about those features expressed quantitatively. The implications of uncertainty are recognized, particularly with reference to the choice of optimal operating conditions. A quantitative estimate of the expected cost of uncertainty, the value of removing all uncertainty from the decision, is made.

An analysis of the butadiene section of a synthetic rubber plant was used to illustrate the procedure. A Nagiev model of the process was readily

optimized with the help of nonlinear pattern search and linear programming. Critical uncertain parameters were identified and then treated as random variables characterized by subjective probability distributions. A set of guidelines for process operation in the face of uncertainty were developed to replace a rigid specification of operating conditions. Monte Carlo simulation was used to estimate the expected cost of uncertainty.

ACKNOWLEDGEMENTS

I wish to thank Dr. D. Quon for his unfailing guidance, helpful criticism and inexhaustable patience throughout our association, and the National Research Council of Canada and the Department of Chemical Engineering for their financial assistance.

Thanks are due to my husband for his understanding and support, both moral and financial, and to Dr. W. Nader for his encouragement.

My thanks are also extended to the many people who were instrumental in the final preparation of the thesis, particularly to (Mrs.) Shirley Morihira, Janice Biswanger, and Mr. and Mrs. Tony Braun.

TABLE OF CONTENTS

APPENDIX G - Sensitivity Analyses

APPENDIX H - Expected Cost of Uncertainty Estimation

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

Since the Industrial Revolution, man has become increasingly dependent on the development and implementation of a rapidly expanding technology to meet his needs and fulfill his desires. The result has been the evolution of the modern industrial nation. Because a major sector of the nation's economy is supported by industry based on some type of processing, the efficient design and operation of processing systems is vital.

Rudd and Watson, in their text (1), have discussed many of the problems encountered in a process engineering study. They define a processing system as a "collection of equipment which effects the transformation of materials through chemical reaction, phase transition, heating and cooling, agglomeration, size reduction, separation, extraction, combustion and so forth". Processing systems range in size and complexity from the basement still to the large commercial systems. The smaller systems are relatively simple and easily exploited. However, as size and complexity increase, more extensive analysis is required to achieve an effective system.

The larger system is more complicated internally and subject to numerous external restrictions. Resources are limited and investment considerable. The system must satisfy many, often conflicting, objective criteria. It is no longer sufficient to accept any system which produces at a profit; some effort must be made to find the system which best achieves the objectives set for it. This latter is the province of optimization theory. An excellent discussion of the organization, techniques and practical aspects of optimization studies, with particular reference to process optimization studies, can be found in Beveridge and Schechter's recent text on the subject (2).

This work is primarily concerned with the optimal operation of existing chemical processing systems. When the system exists, the equipment characteristics and configuration are already set; the investment is fixed. To some extent, the operation of the system has been observed and is known. Solving this process optimization problem implies choosing a set of operating variables so as to optimize some selection criterion while satisfying all internal and external constraints on the system. A comparable

problem arises in the more complex system design studies. A similar analysis is carried out during evaluation of alternate configurations and levels of investment. It is complicated by the need to determine optimal design variables as well as optimal operating variables. Sargent (3) has done a survey and discussion to techniques available for integrated design and optimization of processes. Many of these are applicable to process operation problems as well.

A process optimization study of an existing processing system seeks to improve the operation of the system. The first step in the analysis is the definition of the objective criterion. Most often the intention is to maximize profit and minimize capital investment while meeting all restrictions on the process. When the system exists, investment value is fixed and can be omitted if relative values are used. Sometimes technical criteria are used; normally these can be reduced to some form of economic criteria.

For optimization to take place, the objective criterion must be expressed as a function of the controllable system variables. This step is facilitated by preparing an adequate representation of the system.

The overall structure of the system can be

represented by a block diagram or flowsheet. Each block or process unit represents a portion of the process with one or more specific functions, varying in scope from a distillation tray to an oil refinery, depending on the particular optimization problem. The system consists of a set of process units, interconnected by process streams, which receives input streams and produces output streams.

To evaluate the objective function, the states of the input, output and interlinking streams, and the behavior of the process units, must be defined; the system variables must be specified. Because the variables are inter-related, they cannot all be specified independently,and a model of the behavior of the system, a definition of the relationship between system variables, is required.

The behavior of each process unit is represented by a model. It can be either analytical or empirical; it can be derived from theoretical considerations, or from observed behavior. The model takes the form of a transformation equation which expresses, unambiguously, the unit output variables in terms of unit input variables and process unit characteristics. When the

system exists, most process unit characteristics are defined though a few might remain variable.

The system model is composed of all the individual unit models. With it, the dependent system variables can be determined once the controllable, or decision variables are specified. Because the transformation equations are, in general, difficult to invert, the decision variables are usually chosen from the set of input variables and variable process characteristics.

A processing system is not independent of external influences which can either specify the value or limit the range of variation of the system variables. Restrictions on the design and operation of a processing system arising from external considerations include feedstock and product availability and quality, minimum profit and maximum investment levels, and limits on the production, composition, and disposal of waste products. The system is also subject to restrictions which arise from internal considerations. These are usually constraints on process unit design and operation and include unit dimensions and capacity, acceptable ranges of operating conditions such as flow rates and temperatures, and mass and energy conservation requirements.

For existing systems, these constraints restrict the behavior of the system to feasible modes of operation. The system is also subject to constraints arising from the system modelling procedure. The available data or the form of the model used can restrict the validity of the model to a limited region. These restrictions must be considered in the optimization study to ensure useful results.

Once the system model has been formulated, and the restrictions on system variables identified, the objective function can be finalized. The objective criterion is broken down into components and each component is expressed in terms of technical data, the system variables, available from the system model. This step can be rather complex, particularly in the case of the economic criterion which includes capital costs and the time value of money. For existing systems, the economic objective function is simpler to formulate; only incremental costs need be included.

The optimization model consists of an objective function expressed in terms of system variables, a system model which defines the relationship between the system variables, and a set of restrictions which limits the acceptable range of those variables. Care must be taken

throughout the development of the model to ensure that the model does indeed represent the system.

Once the optimization model has been formulated, only a few more steps prior to the actual optimization remain: derivation of an acceptable solution procedure, simplification of the model if necessary, and verification of the model.

Optimization cannot proceed unless acceptable solutions to the model equations can be found with reasonable ease. When obtaining a solution is inordinately difficult or time-consuming, an unwarranted effort might be required for optimization. If no acceptable solution procedure can be found, the optimization model must be simplified or the project abandoned.

Simplification can be accomplished in several ways. The number of variables in the problem can be reduced by treating those which have a negligible effect on the objective function value as fixed constants. In some cases, the number of equations can be reduced by solving the problem in stages. It may also be possible to reformulate the model and reduce the problem to one of the special cases for which efficient solution procedures exist.

Whatever optimization model is finally adopted,

if the realism of the model is not preserved, optimization would be worthless. Before optimization begins, the optimization model should be verified as well as is possible by comparing the observed or expected behavior of the system with the behavior predicted by the model.

Given a suitable optimization model, optimal operating conditions for the processing system can be found with the help of an appropriate optimization technique. The results of the optimization study will not only indicate optimal operating conditions for the processing system, but also provide insight into any weakness of the optimization model. If necessary, revisions can be made and optimization repeated until acceptable results are obtained.

Optimization theory has been most effective for the design and operation of individual components of processing systems. The optimization model for components is relatively simple, having few variables and few equations. A solution is usually easily obtained, and several effective techniques are available for optimization (2).

Unfortunately, optimization of large processing systems is not as readily accomplished. Developing a

system model from a detailed examination of each process unit results in a large, often nonlinear, system of equations in many variables. Computer simulations such as PACER (4), are typical. Process optimization based on such a model is complicated, if not impossible. The solution of a large nonlinear programming problem is required; no generally applicable approach exists. Most of the available techniques for optimization of large systems (5) are based on decomposition of the problem into several smaller subproblems, taking advantage of the structure of the optimization model. Before one of these methods can be applied, the optimization model must have the appropriate form; modifications might be necessary. Case studies illustrating the application of two decomposition methods, partition programming (6) and Lasdon's multilevel technique (7), support this conclusion: considerable computational effort is required for the successful optimization of a large processing system using a detailed model; the procedure is lengthy and expensive, even with the aid of modern computers.

An alternate approach is to develop an overall, or macroscopic, model of the processing system. Two classes of system variables can be distinguished:

1) the strategic variables, which are directly related to overall system behavior (system input and output variables), and process unit interaction, and

2) the tactical variables, which pertain to the behavior of individual process units.

The detailed system model discussed previously incorporates both strategic and tactical variables, employing detailed process unit models. Overall relationships are emphasized in the macroscopic model which is concerned primarily with strategic system variables, employing simpler, less representative, process unit models. Each process unit is treated as a black box, or an input-output device. Process unit outputs are expressed in terms of inputs via conversion relations whose coefficients represent, quantitatively, the existing technology of the unit. The conversion relations can be very simple, even linear, as long as the range of the strategic variables, over which the model is valid, is specified. Linear macroscopic models and linear programming were used by Newby and Deam (8) in their studies of refinery problems. Their suggestion has considerable industrial support; indeed, most petroleum refineries make use of linear models for optimization.

The optimization model based on the macroscopic system model would include an objective function expressed in terms of strategic system variables, the restrictions required to ensure feasible and acceptable system variable values, and the restrictions on system variables required to preserve model validity. Optimization based on such a model could be speedily and inexpensively accomplished. In addition, the results of the study could be used to identify the critical areas of the process, pointing out areas where further enrichment is needed, and to provide a framework for later optimization of process unit operation.

Throughout the foregoing discussion, the tacit assumption has been made that the system model developed accurately represents the behavior of the processing system. In practice, this is seldom the case. There will often be uncertainty about the values of some of the system model parameters, arising from unreliable and/or incomplete data as to system response. The approximations made during modelling process unit response may also contribute to the uncertainty. If the uncertainty is ignored, the operating conditions accepted as best on the basis of the optimization study may be far from optimal. The presence and magnitude of

uncertainties in the system model should be recognized and taken into account before an optimization study is considered complete.

Process optimization under uncertainty can be tackled by "decision analysis" - Howard's (9) term for application of modern decision theory to problems in decision-making under uncertainty. The decision analysis approach is based on the assumption that the uncertain parameters can be treated as random variables with "known" probability distributions. The probability distributions are assigned on the basis of available information, both subjective and objective, and can be thought of as a measure of the state of knowledge about the uncertain parameters.

Howard (10) divides the decision analysis procedure into three phases: deterministic, probabilistic and informational. A decision analysis approach to process optimization under uncertainty can be outlined within his broad framework.

The deterministic phase begins with the construction of the deterministic optimization model: the establishment of relationships describing process behavior, the identification of decision and state variables and any additional restrictions on their value,

and the formulation of an objective function. Model parameters are assigned nominal values and ranges reflecting initial information; decision variables are assigned nominal values on the basis of optimization of the deterministic model. A sensitivity analysis is performed to identify those decision variables and model parameters which have a critical effect on objective function value in the neighborhood of the optimal solution to the deterministic problem. Those variables and parameters showing low sensitivity can be regarded as known for the remainder of the analysis.

The probabilistic phase deals with the effect of uncertainty in critical parameters on the decision problem. Each parameter identified as critical by the sensitivity analysis is assigned a probability distribution consistent with available information. If some of the critical parameters are not independent, their joint probability distributions must be assigned. For the remainder of the analysis, the critical parameters are treated as random variables characterized by their assigned probability distributions.

The decision problem is now stochastic. In general, the risk preference of the decision-maker must be accounted for. Risk preference is encoded in a

utility function which serves to relate monetary value to the value judgement, or utility, of the decision maker. Then, the optimal solution to the stochastic decision problem is chosen according to the decision rule first proposed by von Neumann and Morgenstern (11): maximize the expected utility of the objective function. When the decision maker is indifferent to risk, decisions are based on the expected value of the objective function.

The informational phase is concerned with estimating the value of removing the uncertainty in the model parameters. The estimator used is the expected cost of uncertainty - the expected value of the difference between the value of the strategy chosen as optimal in the probabilistic phase, and the value of the optimal strategy when all parameter values are known with certainty. It provides a measure of the value of obtaining better information about model parameters in order to obtain an improved optimization solution.

The approach to determining optimal operating conditions, outlined in the foregoing pages, is intended to be generally applicable to optimization of existing processing systems, particularly in the first optimiza-

tion stage of analysis. It is discussed in detail in the following pages and illustrated by a case study.

II. THEORY

In general, the commercial chemical plant is a multicomponent, multiphase processing system composed of interconnected process units, each of which performs one or more operations directed towards the conversion of input streams into desired products. Given an existing plant, it is desired to determine a set of those operating conditions directly affecting overall plant operation which is, in some sense, optimal. Briefly, the procedure suggested here is as follows:

1) Develop a simple optimization model, basing it on a macroscopic model of the processing system.

2) System modelling is usually hampered by notoriously inadequate data, resulting in uncertainty about model parameters. Incorporate the critical uncertain parameters in the optimization model as random variables.

3) The resulting programming problem is stochastic. Find an acceptable solution and hence specify 'optimal' operating conditions.

4) Estimate the value of obtaining more information about the crucial uncertain parameters, thus

enabling rational decision-making about model improvement in the hope of improving system operation.

A. The System Model

The transformation equations which constitute the system model are, essentially, the material balance equations for the processing system. Nagiev (12,13,14) suggested basing material balance calculations on a linear input-output representation of process behavior. The result was a simple material balance formulation for the general multicomponent, multiunit process with arbitrary, defined recycling. The linear system of equations he derived is identical to the system of transformation equations in the economic input-output models introduced by Leontief (15). Nagiev's work forms the basis for the development of the proposed model.

The model is restricted to representation of steady-state relationships among process units and between the processing system and its environment. The system variables considered to be important to overall operation in this formulation are the component flowrates in the feed, product and process streams, and the inter-unit recycle fractions. Each process unit is treated as a simple black box, or input-output device,

which alters input streams linearly to produce output streams.

For a processing system consisting of n distinct interconnected process units with m chemical components in the process streams, define:

f(i,k)= mass feed of component i to unit k from outside the system; an external feed stream.

g(i,k)= total mass flow of component i to unit k from process units and external feed streams; the charge or internal feed to unit k.

a(i,j,k)= mass fraction of the total flow of component i to unit j which, after processing, goes to unit k; a Nagiev or recovery factor.

p(i,k)= the flow of component i from unit k which leaves the system as an external product.

d(i,k)= the mass fraction of the total flow of component i to unit k which leaves the system as product; a product recovery factor.

h(i,k)= artificial feed of component i to

unit k introduced to handle production and disappearance due to reaction.

$s(i,q,k)$ = fraction of the mass flow of component q to unit k which appears as component i in unit k due to reaction; a reaction conversion factor. A negative factor indicates disappearance.

Where there is no reaction in unit k, a material balance taken around the entrance to that unit yields, for each component i,

$$g(i,k) = f(i,k) + \sum_{j=1}^{n} a(i,j,k)\, g(i,j) \qquad \text{(II-1)}$$

Reaction is accounted for by the addition of an artificial feed (or vent) stream to the reaction unit to represent production and disappearance of components due to reaction, an approach suggested by Rosen (16). The material balance equation (II-1) becomes

$$g(i,k) = f(i,k) + \sum_{j=1}^{n} a(i,j,k)\left[g(i,j) + h(i,j)\right] \qquad \text{(II-2)}$$

Similarly, the product stream flowrates are defined by

$$p(i,k) = d(i,k)\left[g(i,k) + h(i,k)\right] \qquad \text{(II-3)}$$

Further, the artificial feed, h(i,k), can be expressed in terms of process flowrates, yielding

$$h(i,k) = \sum_{q=1}^{m} s(i,q,k)\, g(q,k) \qquad \text{(II-4)}$$

These three equations, (II-2), (II-3), and (II-4) are the basic material balance equations for the processing system. The systems of n equations which result from writing each equation for all process units are, in matrix-form:

$$\left[\underline{I}-\underline{A}_i^t\right]\underline{g}_i = \underline{f}_i + \underline{A}_i^t\underline{h}_i \qquad \text{(II-5)}$$

where $$\underline{A}_i^t = \begin{bmatrix} a(i,1,1) & a(i,2,1) & \cdots & a(i,n,1) \\ a(i,1,2) & a(i,2,2) & \cdots & a(i,n,2) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a(i,1,n) & a(i,2,n) & \cdots & a(i,n,n) \end{bmatrix}$$

$$\underline{p}_i = \underline{D}_i\left[\underline{g}_i + \underline{h}_i\right] \qquad \text{(II-6)}$$

where

$$\underline{D}_i = \begin{bmatrix} d(i,1) & & & & \\ & d(i,2) & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & d(i,n) \end{bmatrix}$$

and

$$\underline{h}_i = \sum_{q=1}^{m} \underline{S}_{i,q}\underline{g} \qquad \text{(II-7)}$$

where

$$\underline{S}_{i,q} = \begin{bmatrix} s(i,q,1) & & & & \\ & s(i,q,2) & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & s(i,q,n) \end{bmatrix}$$

In the preceeding equations, as in those to follow, the notation for vectors of the system variables f(i,k), g(i,k), p(i,k) and h(i,k) follows the convention:

$$\underline{g}_i = \begin{bmatrix} g(i,1) \\ g(i,2) \\ \cdot \\ \cdot \\ g(i,n) \end{bmatrix}$$

and

$$\underline{g} = \begin{bmatrix} \underline{g}_1 \\ \underline{g}_2 \\ \cdot \\ \cdot \\ \underline{g}_m \end{bmatrix}$$

Writing the material balance equations for all components results in:

$$\left[\underline{I} - \underline{A}^t\right] \underline{g} = \underline{f} + \underline{A}^t \underline{h} \qquad \text{(II-8)}$$

where

$$\underline{A}^t = \begin{bmatrix} \underline{A}_1^t & & & & \\ & \underline{A}_2^t & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \underline{A}_m^t \end{bmatrix}$$

$$\underline{p} = \underline{D}\left[\underline{g} + \underline{h}\right] \tag{II-9}$$

where

$$\underline{D} = \begin{bmatrix} \underline{D}_1 & & & & \\ & \underline{D}_2 & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \underline{D}_m \end{bmatrix}$$

$$\underline{h} = \underline{S}\,\underline{g} \tag{II-10}$$

where

$$S = \begin{bmatrix} \underline{S}_{1,1} & \underline{S}_{1,2} & \cdot\ \cdot\ \cdot & \underline{S}_{1,m} \\ \underline{S}_{2,1} & \underline{S}_{2,2} & \cdot\ \cdot\ \cdot & \underline{S}_{2,m} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \underline{S}_{m,1} & \underline{S}_{m,2} & & \underline{S}_{m,m} \end{bmatrix}$$

Substituting equation (II-10) in equations (II-8) and (II-9) yields

$$\left[\underline{I} - \underline{A}^t - \underline{A}^t\underline{S}\right] \underline{g} = \underline{f} \tag{II-11}$$

$$\underline{p} = \left[\underline{D} + \underline{D}\,\underline{S}\right] \underline{g} \tag{II-12}$$

or, if

$$\underline{B} = \underline{I} - \underline{A}^t - \underline{A}^t\underline{S} \qquad \text{(II-13)}$$

and

$$\underline{T} = \underline{D} + \underline{D}\ \underline{S} \qquad \text{(II-14)}$$

equations (II-11) and (II-12) may be written

$$\underline{B}\ \underline{g} = \underline{f} \qquad \text{(II-15)}$$

$$\underline{p} = \underline{T}\ \underline{g} \qquad \text{(II-16)}$$

Equations (II-15) and (II-16) are general material balance equations for the processing system; they define the relationships between the component flow rates, $\underline{f}$, $\underline{g}$, and $\underline{p}$, and constitute a system model. When the entries in the matrices $\underline{A}$, $\underline{S}$ and $\underline{D}$ are fixed, the equations are linear and easily solved once one of the system flow vectors is specified.

Another factor which is important to overall system behavior, and should be included as a system variable,is inter-unit recycle. Inter-unit recycle can be represented by a separate process unit on the process flowsheet - a simple stream splitter. The

stream splitter is a unit which splits an input stream into two output streams without altering composition; it is characterized by a split factor (or recycle fraction). In the system model presented here, split factors appear as entries in the matrices $\underline{A}$ and $\underline{D}$. When they are variable, the model is nonlinear.

The fractions a(i,j,k), s(i,q,k) and d(i,k) are model parameters which represent the technology of individual process units. They can be estimated, usually from data on system operation, in several ways; Nagiev (14) and Vela (17) have discussed some of these. Whatever procedure is used to arrive at technology parameter values, certain restrictions, arising from the parameter definitions and conservation of matter, must be observed.

1) The recovery factors, a(i,j,k) and d(i,j) are restricted by

$$d(i,j) + \sum_{k=1}^{n} a(i,j,k) \leq 1.0 \qquad \text{(II-17)}$$

$$0.0 \leq d(i,j) \leq 1.0 \qquad \text{(II-18)}$$

$$0.0 \leq a(i,j,k) \leq 1.0 \qquad \text{(II-19)}$$

2) The restrictions on the reaction conversion factors s(i,q,j) are more complex. The following restrictions apply in all cases

$$\sum_{i=1}^{n} s(i,q,j) \leq 0.0 \qquad \text{(II-20)}$$

$$s(i,i,j) \leq 0.0 \qquad \text{(II-21)}$$

If the amount of component i converted is expressed only in terms of the amount of component i present in the input charge stream, then

$$s(i,q,j) \geq 0.0 \quad , \quad i \neq q \qquad \text{(II-22)}$$

and
$$|s(i,q,j)| \leq 1.0 \qquad \text{(II-23)}$$

When it is desirable to express conversion of component i in terms of the input mass of another component, say q_b, then

$$s(i,q_b,j) < 0.0 \quad \text{for some } q_b = i \qquad \text{(II-24)}$$

and the following constraint must be added explicity to the system model.

$$g(i,j) \geq - s(i,q_b,j)\ g(q_b,j) \qquad \text{(II-25)}$$

Because this type of constraint complicates the model, it is preferable to restrict reaction conversion factors by equation (II-22) for all components in all reaction units.

B. Restrictions on System Variables

The range of variation of the system variables is not unlimited. Constraints arise from external requirements and limitations, the characteristics of process equipment, and the system model. Included in these restrictions are factors such as: feedstock availability and composition, process unit capacities and requirements, product demand and specifications, and model validity limits. To preserve the linearity of the model, each constraint is expressed as a requirement on a linear function of the system flow variables. The set of constraints so defined can be written

$$\underline{R}_f \underline{f} + \underline{R}_g \underline{g} + \underline{R}_p \underline{p} \begin{array}{c} \leq \\ = \\ \geq \end{array} \underline{rhs} \qquad \text{(II-26)}$$

where each constraint takes on only one of the signs from the set $\{\leq, =, \geq\}$.

The model formulation also implies certain restrictions which must be included explicitly. The variable split factors are required to be positive fractions. If there are ns variable split factors, this restriction can be written

$$0.0 \leq sf_k \leq 1.0 \quad k=1,2,\ldots ns \tag{II-27}$$

or, in vector form

$$0.0 \leq \underline{sf} \leq 1.0 \tag{II-28}$$

The system flow variables must be positive, that is

$$\underline{f}, \underline{p}, \underline{g} \geq 0.0 \tag{II-29}$$

C. <u>The Objective Function</u>

In the overall optimization of processing system operation, the most common objective criterion is the minimization of net costs (maximization of net profit). Taking care to preserve linearity, the objective criterion can be expressed in terms of system flow variables in the following form.

minimize $z = \underline{c}_f^t \underline{f} + \underline{c}_g^t \underline{g} - \underline{c}_p^t \underline{p}$ (II-30)

The coefficients $\underline{c}_f$, $\underline{c}_g$ and $\underline{c}_p$ are cost coefficients corresponding to feedstock costs, operating costs, and product values respectively. Because fixed costs do not affect the operation of an existing process, these are not included in the cost function.

D. The Deterministic Decision

The deterministic optimization model consists of the basic system model, the additional constraints defining restrictions on system variables, and the objective function. All model parameters are assumed to be known. If the decision variables are taken to be $\underline{sf}$ and one of the system flow vectors, usually $\underline{f}$, the deterministic optimization problem is the following nonlinear programming problem.

Find $\underline{f}$, $\underline{sf}$

so as to

minimize $z = \underline{c}_f^t \underline{f} + \underline{c}_g^t \underline{g} - \underline{c}_p^t \underline{p}$ (II-30)

subject to:

$\underline{B}\ \underline{g} = \underline{f}$ (II-15)

$$\underline{p} = \underline{T}\ \underline{g} \tag{II-16}$$

$$\underline{R}_f\underline{f} + \underline{R}_g\underline{g} + \underline{R}_p\underline{p} \begin{array}{c} \leq \\ = \\ \geq \end{array} \underline{rhs} \tag{II-26}$$

$$0.0 \leq \underline{sf} \leq 1.0 \tag{II-28}$$

$$\underline{f},\ \underline{p},\ \underline{g},\ \geq 0.0 \tag{II-29}$$

The above problem can be considerably reduced in size by elimination of the dependent variables from the objective function and process constraints. Using equations (II-15) and (II-16), $\underline{g}$ and $\underline{p}$ are expressed in terms of $\underline{f}$ and the technology coefficient matrices, $\underline{B}$ and $\underline{T}$.

$$\underline{g} = \underline{B}^{-1}\underline{f} \tag{II-31}$$

$$\underline{p} = \underline{T}\ \underline{g} = \underline{T}\ \underline{B}^{-1}\underline{f} \tag{II-32}$$

Using these relations to eliminate $\underline{p}$ and $\underline{g}$, the reduced problem can be stated.

find $\underline{f}$, $\underline{sf}$

so as to

minimize $$z = \left[\underline{c}_f^t + \underline{c}_g^t \underline{B}^{-1} - \underline{c}_p^t \underline{T}\ \underline{B}^{-1}\right] \underline{f} \qquad \text{(II-33)}$$

subject to

$$\left[\underline{R}_f + \underline{R}_g \underline{B}^{-1} + \underline{R}_p\ \underline{T}\ \underline{B}^{-1}\right] \underline{f} \ \begin{matrix} \leq \\ = \\ \geq \end{matrix}\ \underline{rhs} \qquad \text{(II-34)}$$

$$0.0 \leq \underline{sf} \leq 1.0 \qquad \text{(II-28)}$$

$$\underline{f} \geq 0.0 \qquad \text{(II-29)}$$

When $\underline{sf}$ is also fixed, the optimization problem is linear and can be efficiently solved by using a standard linear programming algorithm in conjunction with the reduced form of the deterministic decision problem.

The general nonlinear programming problem can be solved with the help of separable programming (18). The nonlinear terms in the system model are first converted to separable form by transformation of variables and then approximated by piecewise linear

functions. The original, unreduced form of the problem must be used, and each linear approximation requires the addition of several new variables and constraints to the problem.

When there are few variable split factors an alternate approach is easier to formulate and could be more efficient. The suggestion is to use a search technique to find optimal values for the split factors, employing linear programming and the reduced form of the problem to solve for optimal flow rates at each step. The nonlinear programming problem then has the following form.

Find $\underline{sf}$

so as to

minimize $z = \left[\underline{c}_f^t + \underline{c}_g^t \underline{B}^{-1} - \underline{c}_p^t \underline{T}\ \underline{B}^{-1}\right] \underline{f}$ (II-33)

subject to

$0.0 \leq \underline{sf} \leq 1.0$ (II-28)

$\underline{f} = \underline{f}_s$ chosen so as to

$$\text{minimize} \quad z_s = \left[\underline{c}_f^t + \underline{c}_g^t \underline{B}^{-1} - \underline{c}_p^t \underline{T}\ \underline{B}^{-1}\right] \underline{f}_s \qquad \text{(II-33)}$$

subject to

$$\left[\underline{R}_f + \underline{R}_g \underline{B}^{-1} + \underline{R}_p \underline{T}\ \underline{B}^{-1}\right] \underline{f}_s \begin{array}{c} \leq \\ = \\ \geq \end{array} \underline{rhs} \qquad \text{(II-34)}$$

$$\underline{f}_s \geq 0.0 \qquad \text{(II-29)}$$

The search technique used must be one which does not require explicit gradient information. The pattern search method developed by Hooke and Jeeves (19) appears to be suitable. It is easily programmed (20) and is reliable as a general technique (21).

Neither nonlinear optimization technique guarantees location of global optima; care must be taken to avoid stopping at a local optimum.

E. Sensitivity Analysis

1. Model Parameters

A fundamental assumption in the formulation of the deterministic optimization model was that all system model parameters were known. In practice,

because of inadequate data, they are not known with certainty. When an uncertain parameter is crucial to the model's representation of process behavior, it may have considerable effect on the optimal solution and objective function value. A sensitivity analysis is performed to determine which of the parameters are critical.

Associated with each parameter is a nominal value, the value assigned for the deterministic model, and an interval, the assumed range of the parameter, which reflect initial information about the parameter. If the uncertain parameter is thought of as being a random variable, the nominal value might correspond to the mean or the mode of the distribution, and the range might correspond to the 10 and 90 percent points on the cumulative distribution function. A sensitivity analysis consists of varying each uncertain parameter over its range, or a fixed portion of its range, and observing the resulting change in objective function value relative to some reference point. If some model parameters are not independently defined, their joint sensitivity must be examined.

Demski (22) has discussed some of the theoretical limitations of sensitivity analysis which should be kept under consideration.

1) Parameter deviations can interact; the sensitivity analysis assumes they do not. A joint sensitivity analysis could be used to help take this into account, but such an analysis might require an inordinate effort.

2) A parameter is identified as sensitive only with reference to the existing model structure and parameter estimates, for a particular decision. The sensitive parameter set may not be stable for parameter and structural deviations.

3) In large models, because the number of parameters which can reasonably be examined is limited, a complete analysis is not possible.

4) The results are model dependent; they are applicable to the real system only in so far as the model accurately represents the system.

If these limitations are considered during the interpretation of the results, the following simple procedure is adequate for sensitizing the macroscopic optimization model. Each uncertain parameter is perturbed, in proportion to its range, about its nominal value and the effect on the objective function value observed. The reference

decision is the solution to the deterministic problem defined by using nominal values for model parameters. In the macroscopic system model presented here very few of the model parameters can be varied independently. Equations (II-17) and (II-20) define joint relationships which must be taken into account in the perturbations. In consequence, when a parameter is identified as sensitive, there may actually be other parameters associated with it which contribute to its critical nature. The critical parameters identified by this sensitivity analysis are those sensitive parameters or groups of parameters which, when perturbed, result in an objective function value change greater than some stipulated acceptable variation.

2. Split Factors

Each variable split factor adds to the computational effort required to solve the optimization problem. It would be advantageous to treat them as fixed constants at the level found to be optimal for the deterministic optimization model. This step is justified if it can be shown that such a move would not appreciably affect the optimal solution and objective function value.

A sensitivity analysis aimed at determining which, if any, split factors are not critical is accomplished in two steps.

1) Perturb each split factor and observe the effect on objective function value, as was done for model parameters. Those split factors exhibiting negligible effect on objective function value can be treated as fixed constants.

2) The remaining split factors are critical if their ranges are those assumed during step 1. A simple check can be made. Each model parameter or group of parameters found to be critical is perturbed as in the previous sensitivity analysis, and the optimal solution to the resulting nonlinear programming problem found. If a split factor shows no appreciable variation in its optimal value, it may be treated as a fixed constant.

F. <u>Encoding Uncertainty</u>

Each of the model parameters identified as critical is assigned a probability distribution in accord with the state of knowledge about the parameter. If no more conclusive information is available, opinion analysis can be used to compile appropriate probability

distributions (22). Of course, the distributions assigned should be consistent with the range and nominal value information assumed earlier; if not, the sensitivity of the affected parameters should be re-examined. The critical parameters are treated as random variables characterized by their assigned distributions for the remainder of the analysis.

Because there is uncertainty about some model parameters, the decision problem is stochastic, and there is risk associated with the choice of an operating strategy. However, management usually acts as if it were indifferent to risk at the low levels of investment normally required for process operation. The high risk capital investment has already been made; once the plant is in existence, the primary concern is in maximizing profit. Consequently, the profit for the processing system is an acceptable estimate of management's utility and the optimal strategy can be selected on the basis of maximizing the expected profit, or minimizing the expected cost.

G. The Stochastic Decision

The optimality criterion for the stochastic decision problem is that the expected value of the

cost function be minimized; that is

$$\text{minimize} \quad \mu_z = \underline{c}_f^t \, \underline{\mu}_f + \underline{c}_g^t \, \underline{\mu}_g - \underline{c}_p^t \underline{\mu}_p \qquad \text{(II-35)}$$

where μ_z denotes the expected value of z and the vectors $\underline{\mu}_f$, $\underline{\mu}_g$ and $\underline{\mu}_p$ are the expected values of the flow rate vectors $\underline{f}$, $\underline{g}$ and $\underline{p}$
Because one of the flow rate vectors, say $\underline{f}$, will be a decision vector with a specified value,

$$\underline{\mu}_f = \underline{f} \qquad \text{(II-36)}$$

and hence, the optimality criterion can be rewritten as

$$\text{minimize} \quad \mu_z = \underline{c}_f^t \underline{f} + \underline{c}_g^t \, \underline{\mu}_g - \underline{c}_p^t \, \underline{\mu}_p \qquad \text{(II-37)}$$

Given the optimality criterion, the stochastic decision problem can be stated formally as

find $\underline{f}$, $\underline{sf}$

so as to

$$\text{minimize} \quad \mu_z = \underline{c}_f^t \underline{f} + \underline{c}_g^t \underline{\mu}_g - \underline{c}_p^t \underline{\mu}_p \tag{II-37}$$

subject to

$$\underline{B}\,\underline{g} = \underline{f} \tag{II-15}$$

$$\underline{p} = \underline{T}\,\underline{g} \tag{II-16}$$

$$\underline{R}_f \underline{f} + \underline{R}_g \underline{g} + \underline{R}_p \underline{p} \quad \begin{matrix} \leq \\ = \\ \geq \end{matrix} \quad \underline{rhs} \tag{II-26}$$

$$0.0 \leq \underline{sf} \leq 1.0 \tag{II-28}$$

$$\underline{f},\ \underline{g},\ \underline{p} \geq 0.0 \tag{II-39}$$

The problem is stochastic because some of the entries in the matrices $\underline{B}$ and $\underline{T}$ are random variables. There are few methods available for handling stochastic optimization subject to inequality constraints, even for linear problems (24). The difficulty lies in the selection of an optimal strategy which will remain feasible for all possible constraint sets.

The common industrial approach, when there is uncertainty about system performance, is to provide

slack (or "fat") so that the objectives for the system can be met in spite of unexpected system behavior (25). In terms of the decision problem, the "fat" ensures that a strategy which satisfies the deterministic model constraints will be feasible for the process whatever the actual values of the uncertain parameters.

Chance-constrained programming, developed by Charnes and Cooper (26), is a generalized approach to stochastic programming which permits each inequality constraint to be violated with some preset probability. The chance-constrained formulation admits no general solution procedure for problems with more than one random variable per constraint, even when those constraints are linear (27).

Another approach is to assign costs for constraint violations and minimize the process cost function plus constraint violation costs. When the uncertainties are in the constraint coefficients, this formulation is difficult to solve (again, a solution can be found only for special cases), and the optimal solution found is dependent on the penalties assigned for constraint violations (18).

In the absence of an acceptable analytical technique, when the random elements are important,

either the unmanageable aspects of the constraints are ignored, or analytical techniques are abandoned, and replaced by simulation of alternatives. Neither approach guarantees an optimal solution to the stochastic decision problem, but information about the effect of uncertainty on the choice of strategy is generated, facilitating the selection of an operating strategy which could be better, and certainly is not worse, than the strategy selected on the basis of a deterministic analysis alone.

The simulation of alternatives may be impractical for the stochastic process optimization problems because there are infinitely many alternatives, and, even if only a few are selected, simulation would require an unwarranted amount of computational effort.

The best available approach is to arrive at some compromise optimization model which is amenable to analysis but retains as much of the uncertain character of the problem as possible. If the stochastic decision problem is expressed in the reduced form (i.e. with dependent variables eliminated from the objective function), the repercussions of uncertainty are more obvious. Because the transformation equations (II-15) and (II-16) express $\underline{f}$ and $\underline{p}$ as functions of $\underline{g}$

it is convenient to treat $\underline{g}$ as a decision vector rather than $\underline{f}$, and

$$\underline{\mu}_g = \underline{g} \qquad \text{(II-38)}$$

By taking expected values on both sides of (II-15) and (II-16),

$$\underline{\mu}_f = E(\underline{B}\ \underline{g}) = E(\underline{B})\ \underline{g} \qquad \text{(II-39)}$$

$$\underline{\mu}_p = E(\underline{T}\ \underline{g}) = E(\underline{T})\ \underline{g} \qquad \text{(II-40)}$$

where the notation E(.) denotes an expected value then, eliminating $\underline{\mu}_f$, $\underline{\mu}_p$, $\underline{f}$ and $\underline{p}$ using (II-39),(II-40), (II-15) and (II-16), the reduced stochastic programming problem can be written

find $\underline{g}$, $\underline{sf}$

so as to

minimize $\quad \mu_z = \left[\underline{c}_f^t E(\underline{B}) + \underline{c}_g^t - \underline{c}_p^t E(\underline{T})\right] \underline{g} \qquad \text{(II-41)}$

subject to

$$\left[\underline{R}_f\underline{B} + \underline{R}_g + \underline{R}_p\underline{T}\right] \underline{g} \ \begin{matrix} \leq \\ = \\ \geq \end{matrix} \ \underline{rhs} \qquad \text{(II-42)}$$

$$0.0 \leq \underline{sf} \leq 1.0 \tag{II-28}$$

$$\underline{g} \geq 0.0 \tag{II-29}$$

Because $\underline{B}$ and $\underline{T}$ have some random elements it is difficult to satisfy the contraints (II-42), particularly the equality constraints. These constraints cannot be ignored completely; without them $\underline{g}$ is unbounded and the solution is trivial. One form of the bounds defined by the constraints (II-42) can be retained if the decision maker is willing to modify his restrictions so that only the expected values of the system flow variables need satisfy those constraints. With this modification, the stochastic decision problem becomes

find $\underline{g}$, $\underline{sf}$

so as to

$$\text{minimize} \quad \mu_z = \left[\underline{c}_f^t E(\underline{B}) + \underline{c}_g^t - \underline{c}_p^t E(\underline{T})\right]\underline{g} \tag{II-48}$$

subject to

$$\left[\underline{R}_f E(\underline{B}) + \underline{R}_g + \underline{R}_p E(\underline{T})\right] \underline{g} \begin{matrix} \leq \\ = \\ \geq \end{matrix} \underline{rhs} \tag{II-43}$$

$$0.0 \leq \underline{sf} \leq 1.0 \qquad \text{(II-28)}$$

$$\underline{g} \geq 0.0 \qquad \text{(II-29)}$$

Recalling that the expected value of a matrix is the matrix of the expected values of its elements, it can be seen that the modified stochastic decision problem is identical in form to the deterministic decision problem. If the nominal values assigned uncertain parameters correspond to the expected values of the uncertain parameters, the decision problems are identical and a solution has already been obtained. If not, (for example, when the nominal values correspond to modes of parameters), the problem is easily solved by methods already discussed for the deterministic decision problem.

H. The "Optimal" Policy

Though the stochastic decision problem has proved to be intractable, an acceptable solution - the optimal solution to the modified problem just discussed - can be found. The next step is to interpret that solution and establish an operating strategy for the processing system.

The standard procedure would be to fix the controllable system variables at the levels indicated by the optimal solution to the modified stochastic decision problem. Such a strategy, S_d, could violate the process restrictions (II-26) if the model parameters had other than their expected values. In that case, the strategy would not be theoretically feasible though, in practice, it might well be acceptable because of "over-design" or "fat" in the process units.

An alternate procedure is to formulate an operating strategy S_s in terms of a set of simple guidelines for step-by-step determination of decision variables once process parameters are known. The guidelines attempt to reproduce the essential characteristics of the optimal solution to the modified problem while avoiding a rigid specification of decision variables. Flow rates which are at their bounds in the optimal solution should be kept there if possible. Split factors should have their optimal solution value. In practice, the guidelines are easily implemented and the resulting operating conditions do not violate process restrictions.

The strategy S_s has several advantages over a simple specification of system variables. If the

guidelines are properly defined, the strategy is always feasible, though not necessarily optimal. Because it allows for variation in system parameters, it is more versatile, and intuitively more appropriate for system operation in the face of uncertainty. Though it has not been shown here it should be possible to approximate more closely the optimal strategy for the stochastic problem, given skillfully defined guidelines. For these reasons, the strategy S_s is a more satisfactory strategy for system operation in the face of uncertainty and, for the purposes of the limited analysis being considered here, is the optimal policy.

J. Cost of Uncertainty

The preceding analysis yielded a strategy S_s, not necessarily optimal, for the operation of the processing system in the face of uncertainty. The objective function value z_s associated with the strategy S_s is a random variable, dependent on the uncertain value of critical system parameters. For a given set of critical system parameters, the lower limit on z_s is z_c, the value of the objective function associated with the optimal operating strategy S_c for the processing system when there is no uncertainty. S_c and z_c are just the optimal operating strategy and the minimum of

the objective function value for the deterministic decision problem when the model parameters are known with certainty. Like z_s, z_c is a random variable.

The purpose of the present analysis is to determine the value of obtaining complete information about the critical parameters - the maximum value of improving the strategy for system operation. This value, the cost of uncertainty, is the difference z_s - z_c, a random variable; until the values of the critical parameters are known it cannot be computed. Without additional information, the best indication of the value of complete information is the expected cost of uncertainty , defined by

$$E(z_s - z_c) = \mu_{zs} - \mu_{zc} \tag{II-44}$$

where μ_{zs} and μ_{zc} are expected values

of z_s and z_c respectively.

Neither parameter can be calculated directly. However, Monte Carlo simulation (28, 29) can be used to obtain estimates. The general procedure is to sample repetitively from the distributions of the critical

parameters, using a random number generator to generate random sets of parameters. For each set, z_s and z_c are calculated. The arithmetic means of the resulting random samples of z_s and z_c are the statistics used to estimate μ_{zs} and μ_{zc}. If N samples were taken, the mean values,

$\bar{z}_s$ and $\bar{z}_c$ are given by

$$\bar{z}_s = \frac{1}{N} \sum_{i=1}^{N} z_s \{i\} \tag{II-45}$$

$$\bar{z}_s = \frac{1}{N} \sum_{i=1}^{N} z_c \{i\} \tag{II-46}$$

$$\text{and } M(z_s - z_c) = z_s - z_c \tag{II-47}$$

where the notation M(.) indicates the arithmetic mean of the contents of the brackets. When using Monte Carlo methods for comparative simulation, Hammersley and Handscomb (29) suggested that the use of the same random numbers in making two unbiased estimates, (e.g. $\bar{z}_s$ and $\bar{z}_c$) results in greater precision for the estimated difference.

The number of samples, N , is chosen so that the variance of the sample mean falls within specified limits, enabling a control on the accuracy of the estimate. Details of the calculation appear in Appendix A, and are based on the work of Shreider (28), and Hadley (30).

III. AN APPLICATION

POLYMER CORPORATION'S BUTADIENE AREA

A. The Process

The approach to process optimization presented in the preceding chapter has been used in a study of a butadiene process similar to that found in Polymer Corporation's Butadiene Area. Typical data on costs and process characteristics, supplied by Polymer Corporation, have been modified to preserve confidentiality but remain representative of butadiene plant operation in the early 1960's.

A flowchart of the butadiene process studied here appears in figure 1; the symbols used are explained in table 1. The process is relatively complicated, consisting of 19 interconnected units, and involving 4 principle components in the process stream: normal butylene, isobutylene, butane and butadiene. Units 1 to 10 are true process units while units 11 to 19 are simple stream splitters. There are 7 external feed streams, pr(i,1) to pr(i,6) and fx(8) to fx(11).

The process splits naturally into two sections, each producing one of the two main products of the butadiene area - butadiene and butyl rubber. Units

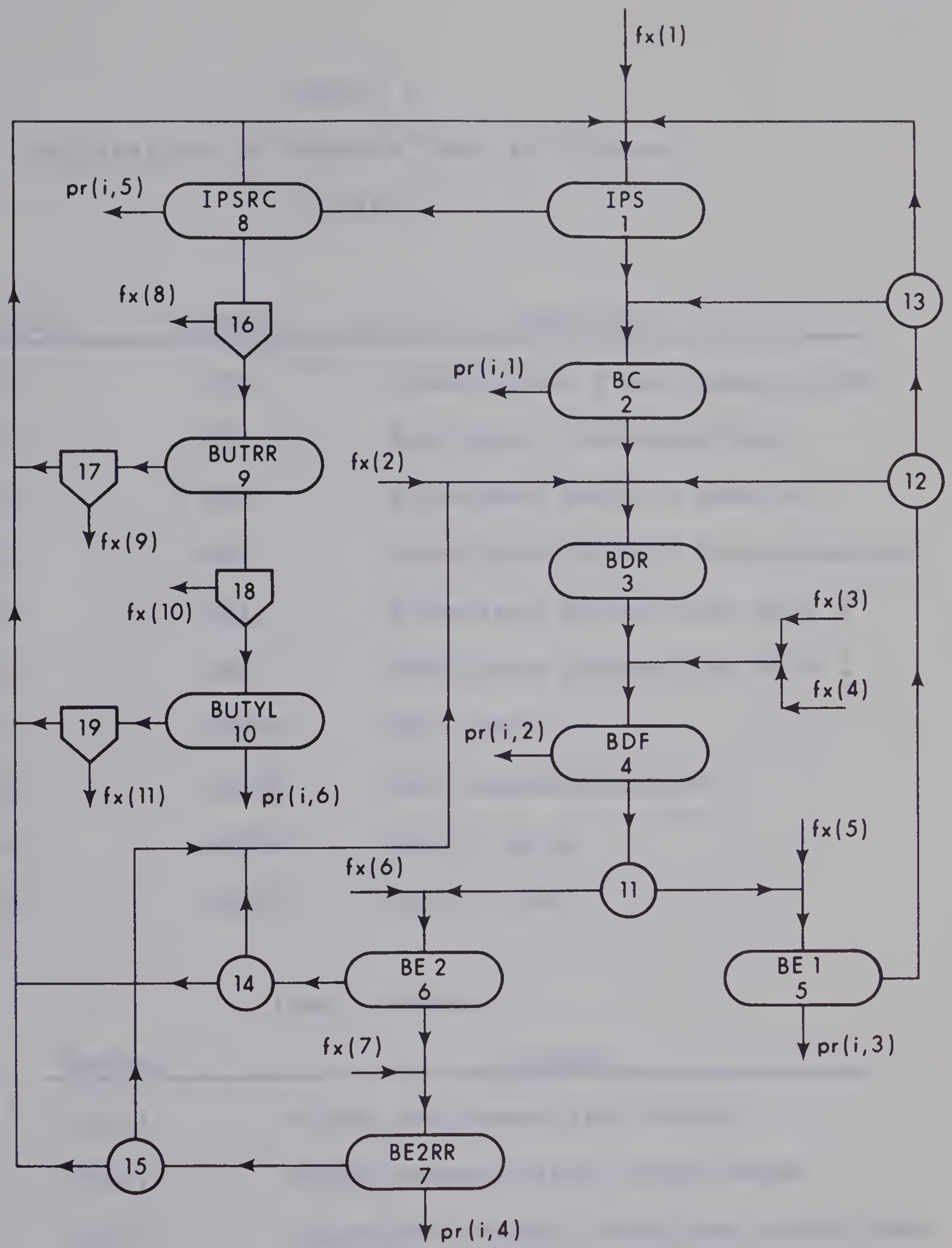

FIGURE 1 BUTADIENE AREA

TABLE I

Explanation of Symbols Used in Figure 1

Units

Unit No.	Symbol	Function
1	IPS	Isobutylene Plant Separation
2	BC	Butylene Concentration
3	BDR	Butadiene Dehydro Reactor
4	BDF	Butadiene Dehydro Fractionation
5	BE1	Butadiene Extraction Unit 1
6	BE2	Butadiene Extraction Unit 2
7	BE2RR	BE2 Rerun
8	IPSRC	IPS Reconcentration
9	BUTRR	Butyl Rerun
10	BUTYL	Butyl Plant

Feed Streams

Symbol	Content
fx(1)	Mixed butylenes and butane
fx(2)	Mixed concentrated n-butylenes
fx(3)	Purchased dilute butadiene,n-butylene
fx(4)	Copolymer recycle - conc. butadiene
fx(5)	Copolymer recycle - conc. butadiene
fx(6)	Purchased dilute butadiene, butylenes
fx(7)	Purchased concentrated butadiene

TABLE 1 cont'd

Product Streams

Symbol	Content
pr(i,1)	Concentrated butane
pr(i,2)	Light, heavy reaction products
pr(i,3)	Concentrated butadiene
pr(i,4)	Concentrated butadiene
pr(i,5)	Concentrated i-butylene
pr(i,6)	Butyl rubber
fx(8)	Straight run i-butylene from IPSRC
fx(9)	Rerun i-butylene from BUTRR
fx(10)	Bottoms i-butylene from BUTRR
fx(11)	Overhead i-butylene from BUTYL

2 - 7 form the butadiene section; units 8 - 10 form the butyl rubber section. Unit 1 splits the main process feed into the feed streams appropriate to each section. The charge to unit 1, fx(1) plus recycle, is predominantly n-butylene, butane and i-butylene. Most of the i-butylene is removed for butyl rubber production while the remaining stream, rich in n-butylene, goes to the butadiene section.

The first unit in the butadiene section, unit 2, concentrates the n-butylene stream by removing most of the butane as product pr(i,1). The concentrated n-butylene stream is then fed to unit 3 with the addition of a concentrated n-butylene feed stream, fx(2), and process recycle. There the butylene is dehydrogenated by catalytic cracking to produce butadiene. Unit 4 removes the light and heavy products of the reaction by fractionation as product pr(i,2). Butadiene and n-butylene may be added prior to unit 4 from feed streams fx(3) and fx(4).

The process stream, now composed mainly of butadiene and n-butylene, is split at unit 11 and goes to either unit 5 or units 6 and 7 for butadiene extraction. There is the option of adding concentrated butadiene, fx(5), at unit 5; dilute

butadiene, fx(6), at unit 6; and concentrated butadiene, fx(7), at unit 7. Unit 5 removes concentrated butadiene as product pr(i,3), with the remaining butylene rich stream being recycled to units 1,2, or 3 via stream splitters 12 and 13. Units 6 and 7 successively concentrate the process stream, producing concentrated butadiene as pr(i,4) from unit 7, and recycling n-butylene to units 1 or 3 via stream splitters 14 and 15.

In practice, the process stream to the butyl rubber section is almost pure i-butylene. Unit 8 concentrates the i-butylene feed from unit 1, producing product pr(i,5). Unit 9 further prepares the process stream for processing in unit 10, recycling a portion of the stream to unit 1. Unit 10 polymerizes the isobutylene, producing butyl rubber as product pr(i,6) and recycling the remainder of the process stream to unit 1. Isobutylene may be removed from the process stream for in-plant use via stream splitters 16, 18 and 19 as products fx(8), fx(10) and fx(11), or for sale via stream splitter 17 as product fx(9).

B. The Optimization Model

1. Further Development

The matrix equations which constitute the optimization model are the system model transformation equations:

$$\underline{B}\ \underline{g} = \underline{f} \tag{II-15}$$

$$\underline{p} = \underline{T}\ \underline{g} \tag{II-16}$$

the restrictions:

$$\underline{R}_f\underline{f} + \underline{R}_g\underline{g} + \underline{R}_p\underline{p} \quad \begin{matrix} \leq \\ = \\ \geq \end{matrix} \quad \underline{rhs} \tag{II-26}$$

$$0.0 \leq \underline{sf} \leq 1.0 \tag{II-28}$$

$$\underline{f}, \underline{p}, \underline{g} \geq 0.0 \tag{II-29}$$

and the objective function:

$$\text{minimize} \quad z = \underline{c}_f^t\underline{f} + \underline{c}_g^t\underline{g} - \underline{c}_p^t\underline{p} \tag{II-30}$$

The cost function does not include fixed costs since these do not affect a decision on process operating conditions.

For this process, the compositions of the external feed streams are constant, resulting in the constraint

$$\underline{f} = \underline{ft}\ \underline{Y} \qquad \text{(II-1)}$$

where

ft(k) = total external feed to unit k

y(i,k) = mass fraction of component i in the total external feed to unit k

$$\underline{ft} = \begin{bmatrix} ft(1) \\ ft(2) \\ \cdot \\ \cdot \\ ft(n) \end{bmatrix}$$

$$\underline{Y} = \begin{bmatrix} \underline{Y}_1 & \underline{Y}_2 & \cdot & \cdot & \cdot & \cdot & \cdot & \underline{Y}_m \end{bmatrix}$$

$$\underline{Y}_i = \begin{bmatrix} y(i,1) & & & & \\ & y(i,2) & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & y(i,m) \end{bmatrix}$$

Using (III-1), $\underline{f}$ can be eliminated from the optimization model. The equations (II-15), (II-26) and (II-29) become

$$\underline{B}\ \underline{g} \ = \ \underline{ft}\ \underline{Y} \qquad \text{(III-2)}$$

$$\underline{R}_{ft}\underline{ft} + \underline{R}_g\underline{g} + \underline{R}_p\underline{p} \ = \ \underline{rhs} \qquad \text{(III-3)}$$

$$\underline{ft},\ \ \underline{p},\ \underline{g} \quad 0.0 \qquad \text{(III-4)}$$

and the objective function can be written

$$\text{minimize} \quad z = \underline{c}_{ft}^t\underline{ft} + \underline{c}_g^t\underline{g} - \underline{c}_p^t\underline{p} \qquad \text{(II-5)}$$

$\underline{R}_{ft}$ and $\underline{c}_{ft}$ are the constraint and cost coefficients of the external feed $\underline{f}$, appropriately redefined in terms of $\underline{ft}$. Equations (II-16), (II-28) and (III-2) through (III-5)

make up a general optimization model.

One other modification to the model is desirable. The model as developed, allows for precisely one feed and one product stream per unit. In practice, as can be seen from the flowchart, figure 1, this need not be the case. A more efficient notation can be developed as follows. Define:

fx(j)=j^{th} external feed stream.

y*(i,j,k) = mass fraction of component i in the j^{th} external feed stream, fx(j), a stream which goes to unit k.

y*(i,j,kk) = 0.0 kk=1,....n kk $\neq$ k

pr(i,j) = mass flow of component i in the j^{th} external product stream.

d*(i,j,k) = mass fraction of the total flow of component i to unit k which leaves the system as the j^{th} product pr(i,j).

d*(i,j,kk) = 0.0 kk=1,....n kk = k

nf = number of external feed streams

np = number of external product streams

Then

$$\underline{fx} = \begin{bmatrix} fx(1) \\ fx(2) \\ \cdot \\ \cdot \\ fx(nf) \end{bmatrix} \tag{II-6}$$

$$\underline{Y}^* = \left[\underline{Y}^*_1 \; . \; . \; . \; \underline{Y}^*_i \; . \; . \; . \; \underline{Y}^*{}_m \right] \tag{II-7}$$

where

$\underline{Y}^*{}_i$ is an (nf x n) matrix

with elements $y^*_{jk} = y^*(i,j,k)$

$$\underline{pr} = \begin{bmatrix} pr(1,1) \\ pr(1,2) \\ \cdot \\ \cdot \\ pr(m,1) \\ pr(m,2) \\ \cdot \\ \cdot \\ pr(m,np) \end{bmatrix} \tag{II-8}$$

$$\underline{D}^* = \begin{bmatrix} \underline{D}_1^* & & & & \\ & \underline{D}_2^* & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \underline{D}_m^* \end{bmatrix} \qquad \text{(II-9)}$$

$\underline{D}_i^*$ is an (np x n) matrix

with elements $d^*_{jk} = d^*(i,j,k)$

The optimization model, written in terms of these variables, consists of the transformation equations:

$$\underline{B}\ \underline{g} = \underline{fx}\ \underline{Y}^* \qquad \text{(III-10)}$$

$$\underline{pr} = \underline{T}^*\ \underline{g} \qquad \text{(III-11)}$$

where

$$\underline{T}^* = \underline{D}^* + \underline{D}^*\underline{S}$$

the restrictions:

$$\underline{R}_{fx}\underline{fx} + \underline{R}_{g}\underline{g} + \underline{R}_{pr}\underline{pr} \;\begin{matrix}\leq\\=\\\geq\end{matrix}\; \underline{rhs} \qquad \text{(III-12)}$$

$$0.0 \leq \underline{sf} \leq 1.0 \qquad \text{(II-28)}$$

$$\underline{fx},\ \underline{g}, \underline{pr} \geq 0.0 \qquad \text{(III-13)}$$

and the objective function:

$$\text{minimize} \quad z = \underline{c}_{fx}^{t}\underline{fx} + \underline{c}_{g}^{t}\underline{g} - \underline{c}_{pr}^{t}\underline{pr} \qquad \text{(III-14)}$$

There is no change in the form of the optimization model; the theory developed in the previous chapter remains applicable.

2. The Specific Model

The specific model has been developed and tested in the form just discussed. The process can be represented by 19 units, 9 of which are stream splitters, characterized by variable split factors. The streams passing through units 16 to 19 contain only one component, isobutylene. Consequently, those units can be eliminated from the formulation provided that their product streams, fx(8) to fx(11) are regarded as negative feed streams to units 8,9, and 10. For this

model, fx(8) to fx(11) appear as external feed streams with negative compositions.

The data for the model, based on that provided by Polymer Corporation, follows. There are four components, designated as follows:

i	component
1	n-butylene
2	i-butylene
3	butane
4	butadiene

The integer parameters are:

no. of components	m = 4
no. of units	n = 15
no. of external feed streams	nf=11
no. of product streams	np=6
no. of variable split factors	ns=5

Pertinent data for the matrices $\underline{B}$, $\underline{Y}^*$, and $\underline{T}^*$, the a(i,j,k), y*(i,j,k) and d*(i,j,k), are given in tables 2 - 4. The matrix of reaction conversion factors $\underline{S}$ has only two nonzero entries. They describe the conversion of n-butylene to butadiene in unit 3. The nominal values are:

s(1,1,3) = -0.4

$$s(4,1,3) = 0.4$$

With these data, the transformation equations are defined.

The constraints which make up the restrictions (III-12) are the feedstock availability limits:

$$fx(1) \leq 0.96 \qquad \text{(III-15)}$$

$$fx(2) \leq 0.105 \qquad \text{(III-16)}$$

$$fx(3) \leq 0.027 \qquad \text{(III-17)}$$

$$fx(4) + fx(5) \leq 0.014 \qquad \text{(III-18)}$$

$$fx(6) \leq 0.038 \qquad \text{(III-19)}$$

$$fx(7) \leq 0.045 \qquad \text{(III-20)}$$

unit capacity constraints:

$$\sum_{i=1}^{m} g(i,2) \leq 1.05 \qquad \text{(III-21)}$$

$$g(2,1) \leq 0.23 \qquad \text{(III-22)}$$

$$g(1,2) \leq 0.66 \qquad \text{(III-23)}$$

$$g(1,3) \leq 1.15 \qquad \text{(III-24)}$$

$$g(4,5) \leq 0.26 \qquad \text{(III-25)}$$

$$g(4,6) \leq 0.26 \qquad \text{(III-26)}$$

$$g(2,10) \leq 0.14 \qquad \text{(III-27)}$$

product requirements:

$$p(4,3) + p(4,4) \leq 0.06 \qquad \text{(III-28)}$$

$$p(4,3) + p(4,4) \geq 0.34 \qquad \text{(III-29)}$$

and model validity requirements:

$$fx(8) - a(2,8,9)\, g(2,8) \leq 0.0 \qquad \text{(III-30)}$$

$$fx(9) - a(2,9,1)\, g(2,9) \leq 0.0 \qquad \text{(III-31)}$$

$$fx(10) - a(2,9,10)\, g(2,9) \leq 0.0 \qquad \text{(III-32)}$$

$$fx(11) - a(2,10,1)\, g(2,10) \leq 0.0 \qquad \text{(III-33)}$$

The constraints (III-30) to (III-33) ensure that the material balance on units 16 to 19 is not violated. Constraints (II-28) and (III-13) complete the restrictions on system variables.

The objective function has the form of (III-14) with coefficients as defined in tables 5-7.

The decision variables for the optimization model will be the external feed streams fx(1) to fx(11) and the split factors for units 11 to 15. They correspond to the recovery factors a(i,11,5), a(i,12,3) a(i,13,1), a(i,14,1) and a(i,15,1); initial values for these factors are given in table 3.

All flow rates are in millions of pounds per day, and costs are in tenths of a dollar per pound. Consequently, the objective function is in hundred thousands of dollars per day. It should be emphasized that the objective function value has no

TABLE 2

Feed Stock Mass Fractions

y*(i,j,k)

j	k \ i	1	2	3	4
1	1	0.4	0.25	0.35	0.0
2	3	0.92	0.03	0.04	0.01
3	4	0.6	-	0.1	0.3
4	4	0.08	0.01	0.01	0.9
5	5	0.08	0.01	0.01	0.9
6	6	0.3	0.3	-	0.4
7	7	0.1	-	-	0.9
8	8	-	-1.0	-	-
9	9	-	-1.0	-	-
10	9	-	-1.0	-	-
11	10	-	-1.0	-	-

TABLE 3

Nominal Values of Recovery Factors

a(i,j,k)

j	k \ i	1	2	3	4
1	2	1.0	0.15	1.0	1.0
1	8	-	0.85	-	-
2	3	0.92	0.9	0.15	0.98
3	4	1.0	0.97	0.9	0.9
4	11	0.95	0.95	0.95	0.95
5	12	1.0	1.0	1.0	0.05
6	7	0.2	0.2	-	0.95
6	14	0.8	0.8	1.0	0.05
7	15	1.0	1.0	-	0.05
8	9	-	0.8	-	-
9	1	-	0.1	-	-
9	10	-	0.9	-	-
10	1	-	0.1	-	-
11	5	0.5	0.5	0.5	0.5
11	6	0.5	0.5	0.5	0.5
12	3	0.8	0.8	0.8	0.8
12	13	0.2	0.2	0.2	0.2

TABLE 3 cont'd

j	k \ i	1	2	3	4
13	1	0.9	0.9	0.9	0.9
13	2	0.1	0.1	0.1	0.1
14	1	0.65	0.65	0.65	0.65
14	3	0.35	0.35	0.35	0.35
15	1	0.05	0.05	0.05	0.05
15	3	0.95	0.95	0.95	0.95

TABLE 4

Nominal Values of Product Recovery Factors

d*(i,j,k)

j	k \ i	1	2	3	4
1	2	0.08	0.1	0.85	0.02
2	4	0.05	0.05	0.05	0.05
3	5	-	-	-	0.95
4	7	-	-	1.0	0.95
5	8	-	0.1	-	-
6	10	-	0.85	-	-

TABLE 5

Feed Stock Costs - $/lb x 10^{-1}

$\underline{c}_{fx}$ c(k) = cost of fx(k)

k	c(k)
1	0.02
2	0.03
3	0.048
4	0.1
5	0.1
6	0.038
7	0.11
8	- 0.034
9	- 0.084
10	- 0.033
11	- 0.034

TABLE 6

Operating Costs - \$/lb x 10^{-1}

$\underline{c}_g$ c(i,j) corresponds to g(i,j)

i	j	c(i,j)
2	1	0.032
1	2	0.032
4	4	0.028
4	5	0.0076
4	6	0.0064
4	7	0.0029
2	10	0.052

TABLE 7

Product Prices - \$/lb x 10^{-1}

$\underline{c}_{pr}$ c(i,j) = value of product pr(i,j)

i	j	c(i,j)
3	1	0.029
1	2	0.012
2	2	0.012
3	2	0.012
4	2	0.012
4	3	0.12
4	4	0.12
2	5	0.026
2	6	0.3

TABLE 8

Linear Problem Solution

Initial Data

Objective Function Value = - 0.2086

Decision Variable	Value
fx(1)	0.736
fx(2)	0.105
fx(3)	0.0
fx(4)	0.0
fx(5)	0.014
fx(6)	0.038
fx(7)	0.0
fx(8)	0.0
fx(9)	0.016
fx(10)	0.0
fx(11)	0.007

a(i,11,5) = 0.5

a(i,12,3) = 0.8

a(i,13,1) = 0.9

a(i,14,1) = 0.65

a(i,15,1) = 0.05

physical significance because fixed costs are not included in the analysis. It is only useful for comparison.

3. Verification

For testing purposes, linear programming was used to find optimal solutions to the specific optimization model for the butadiene process. The split factors were treated as fixed at their initial values as presented in table 2.

A Fortran program was written to generate both the general and reduced forms of the optimization model in a form suitable for input to MPS/360, IBM's mathematical programming system (30). MPS/360 was used to obtain an optimal solution to the linear programming problem, and their parametric procedures were helpful in pinpointing data inconsistencies.

Several data errors and model errors were corrected resulting in the model just presented. The solution to the linear programming problem places the system variables at levels very similar to the original plant operating conditions. It provides an excellent reference point for comparison with the nonlinear optimization procedures to be discussed in later sections. The solution appears in table 8.

The Fortran programs for model generation from data and MPS/360 programs for the linear programming solutions appear in Appendix B along with program documentation. A separate subroutine was written to handle model data input. Printouts, documentation and input data appear in Appendix C. This subroutine was used for model data input to all of the programs used in the course of the butadiene area optimization.

C. The Deterministic Decision

The deterministic decision problem can be defined symbolically as the following nonlinear programming problem

Find $\underline{f}$, $\underline{sf}$

so as to

$$\text{minimize} \quad z = \underline{c}_{fx}^{t}\, \underline{fx} + \underline{c}_{g}^{t}\underline{g} + \underline{c}_{pr}^{t}\underline{pr} \tag{III-14}$$

Subject to:

$$\underline{B}\ \underline{g} = \underline{fx}\ \underline{Y}^{*} \tag{III-10}$$

$$\underline{pr} = \underline{T}^{*}\underline{g} \tag{III-11}$$

$$\underline{R}_{fx}\,\underline{fx} + \underline{R}_{g}\underline{g} + \underline{R}_{pr}\underline{pr} \ \begin{matrix} \le \\ = \\ \ge \end{matrix}\ \underline{rhs} \tag{III-12}$$

$$0.0 \leq \underline{sf} \leq 1.0 \qquad \text{(III-28)}$$

$$\underline{fx},\ \underline{g},\ \underline{pr} \geq 0.0 \qquad \text{(III-13)}$$

Since there are only 5 variable split factors the hill-climbing technique suggested in Chapter II, section D, can be employed to solve the problem. The pattern search method of Hooke and Jeeves (19), as described by Wilde and Beightler (20), is used for the nonlinear search. At each trial, a set of values for the $\underline{sf}$ is generated and used to define a corresponding linear programming problem which is solved to determine optimal external feed rates for that trial. Dantzig's two-phase simplex algorithm (25) can be used to solve the linear programming problem.

The form of the nonlinear programming problem used for the pattern search solution can be described symbolically as:

find $\underline{sf}$

so as to

$$\text{minimize } z = \underline{c}_{fx}^{t}\,\underline{fx} + \underline{c}_{g}^{t}\,\underline{B}^{-1}\,\underline{fx}\ \underline{Y}^{*} - \underline{c}_{pr}^{t}\,\underline{T}^{*}\underline{B}^{-1}\,\underline{fx}\ \underline{Y}^{*} \qquad \text{(III-34)}$$

subject to:

$$0.0 \leq \underline{sf} \geq 1.0 \qquad \text{(III-28)}$$

$\underline{fx} = \underline{fx}_s$ chosen so as to

$$\text{minimize } z_s = \underline{c}^t_{fx}\underline{fx}_s + \underline{c}_g\underline{B}^{-1}\underline{fx}_s\underline{Y}^* - \underline{c}^t_{pr}\underline{T}^*\underline{B}^{-1}\underline{fx}_s\underline{Y}^* \qquad \text{(III-34)}$$

subject to:

$$\underline{R}_{fx}\underline{fx}_s + \underline{R}_g\underline{B}^{-1}\underline{fx}_s\underline{Y}^* - \underline{R}_{pr}\underline{T}^*\underline{B}^{-1}\underline{fx}_s\underline{Y}^* \begin{matrix}\leq\\=\\\geq\end{matrix} \underline{rhs} \qquad \text{(III-35)}$$

$$\underline{fx}_s \geq 0.0 \qquad \text{(III-13)}$$

Note that the reduced form of the problem has been developed, as in chapter 2, by using the transformation equations (III-10) and (III-11) to eliminate the variables $\underline{pr}$ and $\underline{g}$.

The deterministic decision problem was solved in the manner proposed above. The recovery factors, product recovery factors and reaction conversion factors were assigned their nominal values. Initial values for the split factors were the nominal values given in table 3. No computational difficulties were encountered.

The solution to the deterministic problem appears in table 9. As can be seen, the use of nonlinear programming to determine optimal split

TABLE 9.

Deterministic Decision Problem Solution

THIS IS THE OPTIMAL SOLUTION

OBJECTIVE FUNCTION = -0.2681E-01

VARIABLE NAME	VALUE
FX 1	0.82432
FX 2	0.10491
FX 3	0.0
FX 4	0.0
FX 5	0.0
FX 6	0.03800
FX 7	0.0
FX 8	0.0
FX 9	0.01556
FX 10	0.0
FX 11	0.0
A(I,11, 5)	0.74699
A(I,12, 3)	1.00000
A(I,13, 1)	1.00000
A(I,14, 1)	0.09000
A(I,15, 1)	0.0

INTERNAL STREAM - G(I,J)

J I =	1	2	3	4
1	0.34046	0.22876	0.29423	0.00046
2	0.34046	0.03431	0.29423	0.00046
3	0.95443	0.46815	0.29386	0.02508
4	0.57266	0.45411	0.26447	0.36616
5	0.40638	0.32226	0.18768	0.25985
6	0.14904	0.12055	0.06357	0.10321
7	0.02981	0.02411	0.0	0.09805
8	0.0	0.19444	0.0	0.0
9	0.0	0.15556	0.0	0.0
10	0.0	0.14000	0.0	0.0
11	0.54403	0.43140	0.25125	0.34785
12	0.40638	0.32226	0.18768	0.01299
13	0.0	0.0	0.0	0.0
14	0.11923	0.09644	0.06357	0.00516
15	0.02981	0.02411	0.0	0.00588

PRODUCT STREAMS - P(I,J)

J I =	1	2	3	4
1	0.02724	0.00343	0.25010	0.00001
2	0.02863	0.02271	0.01322	0.01831
3	0.0	0.0	0.0	0.24685
4	0.0	0.0	0.0	0.09315
5	0.0	0.01944	0.0	0.0
6	0.0	0.11900	0.0	0.0

factors has brought about a substantial improvement in the objective function value, close to $6,000 dollars per day. The solution was compared with available information on the operation of the existing system to confirm that the results are realistic. No inconsistencies were observed.

A check was made on the location of the optimum by solving the nonlinear programming problem for several sets of initial split factor values. No alternate optima were found. Initial conditions and optimization results for these cases appear in Appendix F.

The program used to solve the nonlinear programming problem can be conveniently divided into four sections, each of which has been dealt with in a separate appendix. Printouts, documentation and computational details are included for each. The mainline program for the pattern search solution and the pattern search algorithm appear in Appendix F. Model data input routines and formats are covered in Appendix C. Appendix E deals with the fortran programs used to generate the reduced form of the optimization model, given completely specified technology matrices. Finally, Appendix D is concerned with the simple Fortran

subroutines written to set up the optimization model in the form required by Dantzig's two-phase simplex algorithm (25), to solve the linear programming problem, and to organize and printout the results and error messages. The material in Appendices C, D, and E is presented separately because these programs were used repeatedly throughout the analysis.

D. Sensitivity Analysis

1. Model Parameters

A sensitivity analysis was carried out on the system model parameters, a(i,j,k), d*(i,j,k) and s(i,q,k) to determine which are critical to the system representation. Most of the parameters are coupled, either by the equivalent of (II-17)

$$\sum_{kk=1}^{np} d^*(i,kk,j) + \sum_{k=1}^{n} a(i,j,k) \leq 1.0 \qquad \text{(III-36)}$$

or by

$$\sum_{i=1}^{m} s(i,q,j) \leq 0.0 \qquad \text{(II-20)}$$

These relations can be written as equalities by adding a term for losses.

$$\sum_{kk=1}^{np} d^*(i,kk,j) + \sum_{k=1}^{n} a(i,j,k) + \text{losses} = 1.0 \qquad \text{(III-37)}$$

$$\sum_{i=1}^{m} s(i,q,j) + \text{losses} = 0.0 \qquad \text{(III-38)}$$

In general, losses are small. If they are treated as constant, then equations (III-37) and (III-38) can be used to define the joint relationships between coupled parameters. The parameter couplings defined by equations (III-37) and (III-38) are easily identified. Unit 4 is a special case. For this unit, the relationships

$$a(i,4,11) = a(q,4,11) \qquad i=1,.....m \qquad \text{(III-39)}$$

$$d^*(i,2,4) = d^*(q,2,4) \qquad i=1,.....m \qquad \text{(III-40)}$$

also hold and must be accounted for in the sensitivity analysis.

Each parameter has been assigned an interval of uncertainty, symmetrical about the nominal value, which reflects initial information about the range of the parameter. Those parameters with an interval size of 0.0 are considered to be known with certainty.

The parameters, their nominal values and the assigned interval size appear in table 10. Parameter couplings are also indicated there.

The sensitivity analysis is carried out by perturbing a single base parameter, adjusting any coupled parameters according to the appropriate relationship, and observing the effect on objective function value. For each perturbation, the split factors are held constant at their optimal level and the optimal solution to the resulting linear programming problem is found. This is the objective function value used for comparison with the reference value, the value of the optimal solution to the deterministic problem. An attempt to compare objective function values without determining new optimal feed rates for the perturbed problem would be meaningless; the optimal strategy for the deterministic problem will not always be feasible for the perturbed problem.

The computer summary of sensitivity analysis results appears in Table 11. Each perturbation is identified only by a perturbation sequence number. This sequence number is correlated with the base parameter perturbed, the amount of the perturbation, and the total objective function value change over the perturbation interval in Table 12.

For this analysis, those parameters whose perturbation caused a single variation in objective function value greater than 4%, and a total variation

TABLE 10

Parameter Couplings, Nominal Values and Interval Size

Parameter Sensitivity Analysis

Interval centered on Nominal Value

Parameter	Nom. Value	Coupled Parameter	Nom. Value	Interval Size
a(2,1,2)	0.15	a(2,1,8)	0.85	0.1
a(1,2,3)	0.92	d*(1,1,2)	0.08	0.1
a(2,2,3)	0.9	d*(2,1,2)	0.1	0.1
a(3,2,3)	0.15	d*(3,1,2)	0.85	0.1
a(4,2,3)	0.98	d*(4,1,2)	0.02	0.0
a(1,3,4)	1.0	-	-	0.0
a(2,3,4)	0.97	-		0.0
a(3,3,4)	0.90	-		0.1
a(4,3,4)	0.90	-		0.1
a(i,4,11)	0.95	d*(i,2,4)	0.05	0.05
a(1,5,12)	1.0	d*(1,3,5)	0.0	0.0
a(2,5,12)	1.0	d*(2,3,5)	0.0	0.0
a(3,5,12)	1.0	d*(3,3,5)	0.0	0.0

TABLE 10 cont'd

Parameter	Nom. Value	Coupled Parameter	Nom. Value	Interval Size
a(4,5,12)	0.05	d*(4,3,5)	0.95	0.05
a(1,6,7)	0.2	a(1,6,14)	0.8	0.1
a(2,6,7)	0.2	a(2,6,14)	0.8	0.1
a(3,6,7)	0.0	a(3,6,14)	1.0	0.0
a(4,6,7)	0.95	a(4,6,14)	0.05	0.05
a(1,7,15)	1.0	d*(1,4,7)	0.0	0.0
a(2,7,15)	1.0	d*(2,4,7)	0.0	0.0
a(3,7,15)	0.0	d*(3,4,7)	1.0	0.0
a(4,7,15)	0.05	d*(4,4,7)	0.95	0.05
a(2,8,9)	0.8	d*(2,5,8)	0.1	0.1
a(2,9,1)	0.1	a(2,9,10)	0.9	0.1
a(2,10,1)	0.1	d*(2,6,10)	0.85	0.1
s(1,1,3)	-0.4	s(4,1,3)	0.4	0.15

TABLE 11.

Result Printout

BUTADIENE AREA - PARAMETER SENSITIVITY

THE OPTIMAL SOLUTION TO THE DETERMINISTIC PROBLEM IS USED AS THE REFERENCE STRATEGY.

DVAL = REFERENCE OBJECTIVE FUNCTION VALUE
= -0.026815

OVAL = OBJECTIVE FUNCTION VALUE FOR THE PERTURBEB PROBLEM UNDER THE OPTIMAL STRATEGY FOR THAT PROBLEM

NPERT = PERTURBATION NUMBER

NPERT	OVAL	OVAL-DVAL	% CHANGE
1	-0.027351	-0.000536	1.999895
2	-0.026016	0.000799	-2.979108
3	-0.025208	0.001606	-5.990613
4	-0.027466	-0.000652	2.430039
5	-0.026836	-0.000022	0.080383
6	-0.026789	0.000025	-0.094429
7	-0.026479	0.000336	-1.252246
8	-0.027146	-0.000332	1.237185
9	-0.026960	-0.000145	0.540092
10	-0.026735	0.000080	-0.298373
11	-0.027539	-0.000725	2.702823
12	-0.025617	0.001197	-4.465028
13	-0.027592	-0.000777	2.898654
14	-0.025637	0.001177	-4.391148
15	-0.026263	0.000552	-2.059314
16	-0.027103	-0.000289	1.076044
17	-0.026838	-0.000023	0.086663
18	-0.026792	0.000023	-0.086121
19	-0.026796	0.000019	-0.070269
20	-0.026827	-0.000013	0.046874
21	-0.026919	-0.000104	0.387690
22	-0.026618	0.000197	-0.734576
23	-0.026578	0.000237	-0.882492
24	-0.026887	-0.000072	0.269713
25	-0.027331	-0.000516	1.925930
26	-0.025227	0.001587	-5.919580
27	-0.025554	0.001261	-4.701079
28	-0.027245	-0.000430	1.603036
29	-0.024954	0.001861	-6.940846
30	-0.028572	-0.001757	6.551767
31	-0.027052	-0.000237	0.883784
32	-0.026188	0.000627	-2.338125

TABLE 12

Perturbations and Results

Parameter Sensitivity Analysis

Perturbation Number	Base Parameter	Base Perturbation	Total % Change Obj. Function
1,2	a(1,2,3)	± 0.05	5.0
3,4	a(2,1,2)	± 0.05	8.4
5,6	a(2,2,3)	± 0.05	0.17
7,8	a(3,2,3)	± 0.05	2.5
9,10	a(3,3,4)	± 0.05	0.83
11,12	a(4,3,4)	± 0.05	7.2
13,14	a(i,4,11)	± 0.025	7.3
15,16	a(4,5,12)	± 0.025	3.2
17,18	a(1,6,7)	± 0.05	0.18
19,20	a(2,6,7)	± 0.05	0.12
21,22	a(4,6,7)	± 0.025	1.1
23,24	a(4,7,15)	± 0.025	1.1
25,26	a(2,8,9)	± 0.05	7.8
27,28	a(2,9,1)	± 0.05	6.3
29,30	a(2,10,1)	± 0.05	13.5
31,32	s(1,1,3)	± 0.075	3.2

greater than 6%, were identified as critical parameters. From tables 11 and 12, in order of sensitivity, the critical parameters are:

Parameter	Coupled Parameter
a(2,10,1)	d*(2,6,10)
a(2,1,2)	a(2,1,8)
a(2,8,9)	d*(2,5,8)
a(i,4,11)	d*(i,2,4)
a(4,3,4)	
a(2,9,1)	a(2,9,10)

On the basis of this sensitivity analysis it is assumed that uncertainty in the remaining non-critical parameters has a negligible effect on the decision problem. They are treated as if they were known, with certainty, to have their nominal values.

As Demski(22) has pointed out, the set of critical parameters just identified is not necessarily stable over changes in model parameters, model structure or process operating conditions. Their applicability to the existing process is dependent on the realism of the model representation.

Printouts, documentation and computational details of programs written to perform the sensitivity analysis appear in Appendix G.

2. Split Factors

A sensitivity analysis was carried out to identify those split factors which have a critical effect on objective function value. It was accomplished in two stages.

In the first stage, each variable split factor was perturbed in turn about its optimal level, the optimal solution to the corresponding linear programming problem was found, and the change in objective function value observed. The computer summary of the results appears in Table 13. The perturbation sequence number appearing on the computer printout, Table 13, is correlated with the split factor perturbation and the split factor's optimal value in Table 14. Split factor a(i,13,1) was not perturbed because unit 13 is not used in the optimal solution to the deterministic problem.

From Table 13, only split factors a(i,11,5) and a(i,12,3) are noticeably sensitive.

The second stage, termed the range analysis, is performed to determine whether or not there could be an appreciable change in the optimal values of the sensitive split factors due to the uncertainty in the critical parameters. Each critical parameter was

perturbed in turn as in the parameter sensitivity analysis. For each perturbation, the non-sensitive split factors were held constant at their optimal level, and the solution to the corresponding nonlinear programming problem found. The computer summary of results appears in Table 15. The perturbation sequence number is correlated with the critical parameter perturbation in Table 16.

As can be seen from Table 15, no appreciable variation in the optimal value of the sensitive split factors was observed. Therefore, the split factors may reasonably be treated as constants, fixed at their optimal value.

As Demski (22) pointed out, these results are valid only so long as no substantial changes are made in model parameters structure; or operating conditions; they need not be applicable to the existing process if the model's representation is poor.

Printouts, documentation and computational details of programs written to perform the sensitivity analysis appear in Appendix G.

TABLE 13.

Result Printout

BUTADIENE AREA - SPLIT FACTOR SENSITIVITY

THE OPTIMAL SOLUTION TO THE DETERMINISTIC PROBLEM IS USED AS THE REFERENCE STRATEGY.

OVAL = REFERENCE OBJECTIVE FUNCTION VALUE
     = -0.026815

DVAL = OBJECTIVE FUNCTION VALUE FOR THE PERTURBEB PROBLEM UNDER THE OPTIMAL STRATEGY FOR THAT PROBLEM

NPERT = PERTURBATION NUMBER

NPERT	OVAL	OVAL-DVAL	% CHANGE
1	-0.026017	0.000797	-2.973577
2	-0.026746	0.000068	-0.254973
3	-0.025339	0.001476	-5.504771
4	-0.026669	0.000145	-0.541857
5	-0.026423	C.000391	-1.460011
6	-0.025911	0.000904	-3.369992
7	-0.026773	0.000042	-0.157488
8	-0.026759	0.000056	-0.207793
9	-0.026702	0.000113	-0.421852
10	-0.026677	0.000137	-0.511807
11	-0.026789	0.000026	-0.097832
12	-0.026760	0.000055	-0.203709

TABLE 14

Perturbations

Split Factor Sensitivity Analysis

Perturbation Number	Split Factor	Optimal Value	Perturbation
1	a(i,11,5)	0.747	+0.053
2	a(i,11,5)	0.747	-0.047
3	a(i,11,5)	0.747	+0.103
4	a(i,11,5)	0.747	-0.097
5	a(i,12,3)	1.0	-0.05
6	a(i,12,3)	1.0	-0.1
7	a(i,14,1)	0.09	-0.05
8	a(i,14,1)	0.09	+0.05
9	a(i,14,1)	0.09	-0.09
10	a(i,14,1)	0.09	+0.1
11	a(i,15,3)	1.0	-0.05
12	a(i,15,3)	1.0	-0.1

TABLE 15.

Range Analysis - Result Summary

BUTADIENE AREA - S.F. RANGE ANALYSIS

THE OPTIMAL SOLUTICN TO THE DETERMINISTIC PROBLEM IS USED AS THE REFERENCE STRATEGY.

OVAL = REFERENCE OBJECTIVE FUNCTION VALUE
= -0.026805

OVAL = OBJECTIVE FUNCTION VALUE FOR THE PERTURBEB PROBLEM UNDER THE OPTIMAL STRATEGY FOR THAT PROBLEM

NPERT = PERTURBATION NUMBER

% CHANGE = % CHANGE IN OBJECTIVE FUNCTION VALUE

NPERT	% CHANGE	S.F.1	S.F.2
1	-5.98	0.740	1.000
2	2.42	0.740	1.000
3	2.71	0.740	1.000
4	-4.48	0.740	1.000
5	3.06	0.740	0.980
6	-4.41	0.740	1.000
7	1.91	0.740	1.000
8	-5.91	0.740	1.000
9	-4.71	0.740	1.000
10	1.60	0.740	1.000
11	-6.95	0.740	1.000
12	6.56	0.740	1.000

SPLIT FACTOR	MEAN	VARIANCE
1	0.740	0.17E-05
2	0.998	0.34E-04

TABLE 16

Perturbations

Split Factor Range Analysis

Perturbation Number	Critical Parameter	Perturbation
1	a(2,1,2)	+0.05
2	a(2,1,2)	-0.05
3	a(4,3,4)	+0.05
4	a(4,3,4)	-0.05
5	a(i,4,11)	+0.025
6	a(i,4,11)	-0.025
7	a(2,8,9)	+0.05
8	a(2,8,9)	-0.05
9	a(2,9,1)	+0.05
10	a(2,9,1)	-0.05
11	a(2,10,1)	+0.05
12	a(2,10,1)	-0.05

E. The Probabilistic Phase

1. Encoding Uncertainty

Each of the model parameters identified as uncertain is assigned a probability distribution in accordance with the state of knowledge about the parameter. In this case the probability distributions are subjective, assigned on the basis of opinion analysis. The distributions are assumed to be independent and normal with parameters as detailed in Table 17.

The expected value of each distribution is identical to the nominal value assigned to the critical parameter. The probability that the critical parameter value will lie outside the interval assigned for the sensitivity analysis can be calculated from the definition of the normal distribution (30). That is

$$p(|\beta_i - \mu_{\beta_i}|>\delta_{\beta_i}) = 2\,(1-\Phi(\delta_{\beta_i}/\sigma_{\beta_i})) \qquad \text{(III-41)}$$

where $\beta_i = i^{th}$ critical parameter

TABLE 17

Subjective Probability Distribution Parameters

Normal Distribution

μ_β = expected value

σ_β^2 = variance

Symbol	Parameter	μ_β	σ_β^2
β_1	d*(2,6,10)	0.85	0.0004
β_2	a(2,1,8)	0.85	0.0004
β_3	a(2,8,9)	0.8	0.0004
β_4	a(i,4,11)	0.95	0.0001
β_5	a(4,3,4)	0.9	0.0004
β_6	a(2,9,1)	0.1	0.0004

μ_{β_i} = expected value, β_i

$2\delta_{\beta_i}$ = $_i$ interval size

$\sigma^2_{\beta_i}$ = variance, β_i

Φ = standardized cumulative normal probability distribution

p (ε) = probability of the event ε

For all critical parameters, that probability is 0.0124. Because the normal distribution is unbounded there is a finite probability that the critical parameter value, β_i will violate one of the constraints (II-18) and (II-19). It is indistinguishable from 0.0000 and hence negligible. Details of these calculations appear in Appendix A.

The subjective probability distributions have been shown to be consistent with the information available for the sensitivity analysis.

2. The Optimal Policy

Because the expected values of the critical parameter distributions are identical to their nominal values, the solution to the modified stochastic

problem is identical to the solution to the deterministic problem, Table 9. An optimal policy, expressed in terms of a set of guidelines for process operation, can be formulated on the basis of that solution.

The following characteristics of the optimal solution to the deterministic problem are noted:

fx(6)	= 0.038	upper limit	-	(III-19)
fx(9)	= 0.01556	upper limit	-	(III-31)
fx(3),fx(4),fx(5),fx(7) = 0.0				
fx(8), fx(10),fx(11) = 0.0				
g(2,10)		upper limit	-	(III-27)
p(4,3) + p(4,4)		upper limit	-	(III-29)
g(2,1)	= 0.22876	close to capacity	-	(III-22)
g(4,5)	= 0.25985	close to capacity	-	(III-25)

A set of guidelines formulated on the basis of these characteristics is:

1. Use all available fx(9).
2. Use all available fx(6).
3. Use no fx(8), fx(10), fx(11).
4. Use no fx(3), fx(4), fx(5), fx(7).
5. Use as much fx(1) as possible.
6. Use as much fx(2) as possible.

The guidelines can be formally incorporated into an optimization model by specifying a new objective function to account for guidelines 1,5, and 6 and additional constraints ensuring adherance to guidelines 2,3, and 4. A suitable objective function would take the form:

$$\text{minimize} \quad z_s = c_1 fx(9) + c_2 fx(1) + c_3 fx(2) \qquad \text{(III-42)}$$

$$c_1 < c_2 < c_3$$

Examination of the optimal solution to the deterministic problem reveals other useful information. For the butadiene process discussed here, the butyl rubber section can operate close to capacity. There is excess capacity in the butadiene section, and hence opportunity for additional profit if additional markets and additional suitable feedstock, such as fx(2), can be found.

F. Expected Cost of Uncertainty

The expected cost of uncertainty is an indication of the value of obtaining complete information about the critical parameters. Monte Carlo simulation, as outlined in Chapter II, section J, was

used to obtain an unbiased estimate of the expected cost of uncertainty.

A series of independent random samples of the cost of uncertainty is generated. For the j^{th} sample, the procedure is:

1. Using normally distributed random numbers supplied by IBM's random number generator (32), an independent random, $\beta_i \{j\}$, is obtained from each critical parameter's probability distribution.

2. The j^{th} system model is defined using the $\beta_i \{j\}$. The model is linear because the split factors can be treated as fixed.

3. The objective function value $z_s \{j\}$, corresponding to the operating strategy formulated in terms of guidelines (section E-2) is calculated. The solution of a linear programming problem is required.

4. The objective function value $z_c \{j\}$, corresponding to the optimal strategy for the j^{th} system model, is calculated. The solution of the deterministic optimization problem for the j^{th} system model, a linear programming problem, is required.

5. The cost of uncertainty is $z_s \{j\} - z_c \{j\}$. If N samples were generated an estimate of the expected cost of uncertainty is the arithmetic mean of the random samples, that is:

$$E(z_s - z_c) \simeq M(z_s - z_c) = \bar{z}_s - \bar{z}_c \qquad \text{(II-47)}$$

where

$$\bar{z}_s = \frac{1}{N} \sum_{i=1}^{N} z_s\{i\} \qquad \text{(II-45)}$$

$$\bar{z}_c = \frac{1}{N} \sum_{i=1}^{N} z_c\{i\} \qquad \text{(II-46)}$$

Accuracy is estimated as suggested in Appendix A.

After 200 simulations, the expected cost of uncertainty was estimated to be 130 dollars per day. With a 95% confidence level, the estimate falls within the interval (130 $\pm$ 20) dollars per day. The standard deviation of the cost of uncertainty is about 140 dollars per day. The computer summary of results appears as Table 18.

The cost of uncertainty is somewhat lower than the sensitivity analysis on parameters would indicate. It is just 0.5% of the objective function value, while single critical parameter perturbations resulted in changes greater than 4%. It is unlikely that this is entirely a result of interaction of deviations. A more plausible explanation is that the low cost of uncertainty is an indication of the value of expressing

the optimal policy in terms of a set of guidelines.

A computer printout of sample points is included in Appendix H, table H-4. From the tabulation it can be seen that the cost of uncertainty,(the regret), is usually either negligible or on the order of 300 dollars per day. (Values less than about 1.0×10^{-7} in magnitude are equivalent to zero due to roundoff errors). This indicates that the strategy based on guidelines was near optimal much of the time, but failed to account for all possibilities. The operating strategy might be improved by using detailed information generated for the simulation procedure to develop a more versatile set of guidelines.

Printouts, documentation and computational details of programs written to perform the cost of uncertainty estimation appear in Appendix H.

TABLE 18.

Expected Cost Estimation Results

```
RESULTS OF MONTE CARLO SIMULATICN

NO. OF SAMPLFS TAKEN =  200
NO. OF SAMPLES REJECTED =     0

EXPECTED COST OF UNCERTAINTY =  0.133E-03   VAR. =  0.21E-07

FXPECTED COST, REF. STRATEGY = -0.263E-01 VAR. =  0.17E-05
EXPECTED COST, CERTAINTY = -0.265E-01 VAR. =  0.15E-05

PRECISION, COST OF UNCERTAINTY ESTIMATE =  0.20E-04
CONFIDENCE LEVEL REQUESTED = 0.950

SIMULATICN COMPLETED
```

IV. DISCUSSION AND CONCLUSIONS

The use of a macroscopic system model in conjunction with decision analysis techniques has been suggested as a valid approach to the problem of optimal operation of an existing processing system in the face of uncertainty.

A. The Butadiene Process

The butadiene process case study provided valuable insight into optimal operation of the process with a minimum of effort. Deterministic optimization of the nonlinear process model yielded a set of overall operating conditions which could reduce the marginal cost function for the process by about 6,000 dollars per day. A sensitivity analysis pinpointed those system parameters which are most critical to optimal system operation, and showed that split factors were not critical in the neighborhood of the optimum. A set of guidelines were developed which would indicate near-optimal operating conditions regardless of the exact state of the critical parameters. An estimate of the cost of uncertainty was made, showing that little could be gained by further information gathering. The cost of uncertainty was lower than would have been expected from sensitivity analysis results. This is

an indication of the effect of expressing the strategy for operation in the face of uncertainty in terms of a set of guidelines.

This information forms an excellent base for further, more detailed, optimization studies. The critical areas of the process have been identified, overall operating conditions for individual units have been specified, and any changes in unit behavior can be quickly evaluated.

B. General Application

The approach suggested here is expected to be generally applicable to a wide variety of process operation problems. It has the advantage of providing needed information about overall operation with relatively little effort. Hence, it provides a good framework for later suboptimization.

The system model developed is mathematically tractable and relatively easily defined though in some cases it may not be adequate. It should be most useful for large problems with extensive sections of the process classed as units; the larger units are more amenable to simple representation.

The simple sensitivity analysis for system parameters should be adequate for most applications. In large problems the number of parameters that can

reasonably be sensitized is limited. Prior knowledge of process operation should be sufficient to indicate areas most likely to be critical.

For most industrial problems, it would be unrealistic to act as if uncertainty does not exist. The definition of the optimal strategy in the face of uncertainty in terms of simple guidelines, suitable for actual process operation, holds promise. The approach is more realistic than a straight forward specification of operating conditions, and has the capability of being nearer to optimal.

The estimation of the cost of uncertainty is valuable as a quantitative indication of the additional expenditure warranted in an effort to remove uncertainty from the model. Monte Carlo simulation may appear to be unreasonably expensive for this purpose. This is not necessarily so. As few as 25 - 30 simulations may be adequate, depending on the accuracy required of the estimate.

As with most optimization methods, the major limitation lies in the fact that the results of the study are valid only for the model. They are applicable to the process only to the extent that the model is an accurate representation of process behavior.

C. Future Work

The extension of the present work to dynamic system models would be advantageous. The theory of dynamic input-output models used in economics, has been developed (5). The only problems appear to be handling variable split factors and formulating the appropriate process model.

The extension of the analysis to account for uncertainty in the cost coefficients and constraint requirements is straight forward. Though the stochastic optimization model is more complex, the same approach can be used to handle it.

The macroscopic model could be useful in process design problems, both for evaluation of alternate configurations and determination of operating conditions. It may have some use in setting overall process unit characteristics as well. There, the emphasis would be on determining the best of possible system parameters.

NOMENCLATURE

$a(i,j,k)$	-	recovery factor
$\underline{A}_i^t$	-	recovery factor matrix, i^{th} component
$\underline{A}^t$	-	recovery factor matrix
$\underline{B}$	-	coefficient matrix for internal stream transformation equation
$\underline{c}_f$	-	marginal cost coefficients of component feed streams $\underline{f}$
$\underline{c}_{ft}$	-	marginal cost coefficients of feed streams $\underline{ft}$
$\underline{c}_{fx}$	-	marginal cost coefficients of feed streams $\underline{fx}$
$\underline{c}_g$	-	marginal cost coefficients of internal streams $\underline{g}$
$\underline{c}_p$	-	marginal cost coefficients of product streams $\underline{p}$
$\underline{c}_{pr}$	-	marginal cost coefficients of product streams $\underline{pr}$
c_1, c_2, c_3	-	cost coefficients - guideline optimization model
$d(i,k)$	-	product recovery factor for $p(i,k)$
$\underline{D}_i$	-	product recovery factor matrix for $\underline{p}_i$
$\underline{D}$	-	product recovery factor matrix for $\underline{p}$
$d^*(i,j,k)$	-	product recovery factor for $pr(i,j,k)$
$\underline{D}_i^*$	-	product recovery factor matrix for $\underline{pr}_i$
$\underline{D}$	-	product recovery factor matrix for $\underline{pr}$

E(.)	-	expected value
f(i,k)	-	i^{th} component, external feed stream to unit k
$\underline{f}_i$	-	external feed stream vector of i^{th} component, f(i,k)
$\underline{f}$	-	external feed stream vector of f(i,k)
ft(k)	-	external feed stream to unit k
$\underline{ft}$	-	external feed stream vector of ft(k)
fx(k)	-	k^{th} external feed stream
$\underline{fx}$	-	external feed stream vector of fx(k)
g(i,k)	-	i^{th} component, internal feed stream to unit k
$\underline{g}_i$	-	internal feed stream vector, i^{th} component
$\underline{g}$	-	internal feed stream vector
h(i,k)	-	i^{th} component, artificial feed stream to unit k
$\underline{h}_i$	-	artificial feed stream vector, i^{th} component
$\underline{h}$	-	artificial feed stream vector
$\underline{I}$	-	identity matrix
M(.)	-	arithmetic mean
m	-	no. of components
n	-	no. of units
nf	-	no. of feed streams
np	-	no. of product streams
ns	-	no. of variable split factors

N	-	no. of random samples, Monte Carlo simulation
p(i,k)	-	i^{th} component, external product stream from unit k
$\underline{p}_i$	-	external product stream vector of i^{th} component, p(i,k)
$\underline{p}$	-	external product stream vector of p(i,k)
pr(i,k)	-	i^{th} component, k^{th} external product stream
$\underline{pr}_i$	-	external product stream vector of i^{th} component, pr(i,k)
$\underline{pr}$	-	external product stream vector of pr(i,k)
$\underline{rhs}$	-	requirements vector, restriction relations
$\underline{R}_f$	-	matrix of coefficients of $\underline{f}$, restriction relations
$\underline{R}_{ft}$	-	matrix of coefficients of $\underline{ft}$, restriction relations
$\underline{R}_{fx}$	-	matrix of coefficients of $\underline{fx}$, restriction relations
$\underline{R}_g$	-	matrix of coefficients of $\underline{g}$, restriction relations
$\underline{R}_p$	-	matrix of coefficients of $\underline{p}$, restriction relations
$\underline{R}_{pr}$	-	matrix of coefficients of $\underline{pr}$, restriction relations
s(i,q,k)	-	reaction conversion factor, component i to component q in the k^{th} unit

$\underline{S}_{i,q}$	-	reaction conversion factor matrix, component i to component q
$\underline{S}$	-	reaction conversion factor matrix
sf_k	-	k^{th} variable split factor
$\underline{sf}$	-	vector of variable split factors
$\underline{T}$	-	coefficient matrix for product stream $\underline{p}$ transformation equation
$\underline{T}^*$	-	coefficient matrix for product stream $\underline{pr}$
y(i,k)	-	mass fraction of component i in ft(k)
$\underline{Y}_i$	-	mass fraction matrix for i^{th} component, $\underline{ft}$
$\underline{Y}$	-	mass fraction matrix for $\underline{ft}$
y*(i,j,k)	-	mass fraction of component i in fx(k)
$\underline{Y}_i^*$	-	mass fraction matrix for i^{th} component, $\underline{fx}$
$\underline{Y}^*$	-	mass fraction matrix for $\underline{fx}$
z	-	objective function value
z_c	-	objective function value for S_c
$z_c\{j\}$	-	z_c, j^{th} random sample
$\bar{z}_c$	-	arithmetic mean, z_c, after N random samples
z_s	-	objective function value for S_s
$z_s\{j\}$	-	z_s, j^{th} random sample
$\bar{z}_s$	-	arithmetic mean, z_s,after N random samples
$p(\varepsilon)$	-	probability of event ε
S_c	-	optimal strategy under certainty
S_D	-	deterministic strategy - specification of system variables

S_s	-	stochastic strategy - specification of guidelines
β_i	-	i^{th} critical parameter, treated as a random variable
$\beta_i\{j\}$	-	j^{th} random sample of β_i
σ_{β_i}	-	one half interval size, β_i
$\Phi(t)$	-	standardized cumulative normal probability distribution
μ_{β_i}	-	expected value, β_i
μ_z	-	expected value, z
$\underline{\mu}_f$	-	expected value, $\underline{f}$
$\underline{\mu}_g$	-	expected value, $\underline{g}$
$\underline{\mu}_p$	-	expected value, $\underline{p}$
$\sigma^2_{\beta_i}$	-	variance, β_i

B I B L I O G R A P H Y

1. Rudd, D.F. and Watson, C.C., "Strategy of Process Engineering", Chapter 1, John Wiley & Sons, New York, 1968.

2. Beveridge, G.S. and Schechter, R.S. "Optimization Theory and Practice", McGraw Hill Books Company, 1970.

3. Sargent, R.W.H., "Integrated Design and Optimization of Processes:, Chem.Eng. Prog., 63, no. 9, 71, (1967).

4. Shannon, P.T., Johnson, A.I., Crowe, C.M., Hoffman, T.W. Hamicki, A.E., and Woods, D.R., "Computer Simulation of a Sulfuric Acid Plant", Chem. Eng. Prog., 62, No.6, 49, (1966).

5. Lasdon, L.S., "Optimization Theory for Large Systems", The Macmillan Co., New York, 1970.

6. Ornea, J.C. and Eldredge, G.C., "Nonlinear Partitioned Models for Plant Scheduling and Economic Evaluation", AICLE - I. Chem. E. Symposium Series, no.4, 119, (1965).

7. Brosilow, C. and Nunez E. "Multi-Level Optimization Applied to a Catalytic Cracking Plant", Can. J. Chem. Eng., 46, 205, (1968).

8. Newby, W.J. and Deam, R.J.,"Optimization and Operational Research ", in "Proceedings of Symposium on Process Optimization, London 1962", Inst. Chem. Engrs., London, 14, 1962.

9. Howard, R.A. "Decision Analysis: Applied Decision Theory", in "Proceedings of the Fourth International Conference on Operational Research", Hertz, D.B. and Melese, J., ed, Wiley-Interscience, New York, 55, 1966.

10. Howard, R.A., "The Foundations of Decision Analysis", IEEE Trans. on Systems Science and Cybernetics, 4, no.3, 211, (1968).

11. von Neumann, J. and Morgenstern, O., "Theory of Games and Economic Behavior", Princeton University Press, Princeton, New Jersey,1947.

12. Nagiev, M.F., "Material Balance in Complex and Multistage Recycle Chemical Processes", Chem Eng. Prog., 53, No.6, 297, (1957).

13. Nagiev, M.F., "Theory of Complex Recycling Processes of Petrochemical Synthesis", Fifth World Petroleum Congress, 297, (1959).

14. Nagiev, M.F., "The Theory of Recycle Processes in Chemical Engineering", The MacMillan Co., New York, 1964.

15. Leontief, W.W., "The Structure of the American Economy, 1919 - 1939", 2nd Ed., part II, Oxford University Press, New York, 1951.

16. Rosen, E.M., "A Machine Computation Method for Performing Material Balances", Chem.Eng Prog., 58, no.10, 69, (1962).

17. Vela, M.A., "Use Fractions for Recycle Balances, part II", Petrol. Refiner, 40, no.6, 189, (1961).

18. Hadley, G., "Nonlinear and Dynamic Programming", Chapters 4 and 5, Addison-Wesley Publishing Co. Inc., New York, 1964.

19. Hooke, R.A. and Jeeves,T.A., "'Direct Search' Solution of Numerical and Statistical Problems", J.Assn. Comp. Mach., 8, no.2 212, (1961).

20. Wilde, D.J. and Beightler, C.S., "Foundations of Optimization", Chapter 7, Prentice-Hall Inc. New Jersey, ch.7, 1967.

21. Leon, A., "A Comparison of Eight Known Optimizing Procedures", in "Recent Advances in Optimization Techniques", Lavi, A. and Vogl, T.D., ed., John Wiley and Sons, New York, 1966.

22. Demski, J.S., "Some Considerations in Sensitizing an Optimization Model", J.Ind.Eng., <u>19</u>, no.9, 463, (1968).

23. Reilly, P.M. and Johri, H.P., "Decision-Making Through Opinion Analysis", Chemical Engineering, <u>76</u>, no.7, 122, (1969).

24. Beale, E.M.L., Mathematical Programming in Practice, Chapter 17, John Wiley and Sons, New York, 1968.

25. Dantzig, G.B., "Linear Programming and Extensions", Chapters 5 and 25, Princeton University Press, New Jersey, 1963.

26. Charnes, A and Cooper, W.W., "Chance - Constrained Programming", Management Science, <u>6</u>, 73,(1959).

27. Miller, B.L. and Wagner, H.M., "Chance - Constrained Programming with Joint Constraints",Operations Research, 13, 930, (1965).

28. Shreider, Yu.A., "Method of Statistical Testing", Chapters 1 and 2, Elsevier Publishing Co., Amsterdam, 1964.

29. Hammersley, J.M. and Handscomb, D.C., "Monte Carlo Methods", Chapter 4, Methuen & Co. Ltd., London, 1965.

30. Hadley, G., "Introduction to Probability and Statistical Decision Theory", Chapters 7 and 9, Holden - Day, Inc., San Francisco 1967.

31. "Mathematical Programming System/360 (360A-Co-14x) Linear and Separable Programming - User's Manual" IBM Application Program H20-0476-0, 1967.

32. "System 360 Scientific Subroutines Package (300A-CM-03X) Version II Programmers Manual" IBM Application Program H20-0205-2, 1966.

APPENDIX A

CALCULATIONS

Table of Contents

List of Tables

1. Accuracy of Monte Carlo Estimation of an Expected Value

The following discussion of the accuracy of Monte Carlo estimation of expected value is based on material appearing in the texts of Schreider (28) and Hadley (30).

The statistic used to estimate the expected value, E(x), of the random variable, x, is the arithmetic mean $\bar{x}$ of N independent samples x{j} from the probability distribution of x. The arithmetic mean is defined by

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x\{i\} \qquad \text{(A-1)}$$

From the central limit theorem of probability theory, for large N the distribution of the random variable $\bar{x}$ is approximately normal. The expected value and variance of $\bar{x}$ are give by

$$E(\bar{x}) = E(x) = \mu_x \qquad \text{(A-2)}$$

$$\sigma_{\bar{x}}^2 = \frac{\sigma_x^2}{N} \qquad \text{(A-3)}$$

It is desired to specify that, with probability ,
the estimate $\overline{x}$ should fall within the interval
$(\mu_x \pm \delta)$. From the definition of the normal distribution

$$p(|\overline{x} - \mu_x| > \delta) \simeq \lambda = 2\left[1-\Phi\left(\frac{\delta}{\sigma_{\overline{x}}}\right)\right] \tag{A-4}$$

If

$$\delta = \alpha\sigma_{\overline{x}} \tag{A-5}$$

then, (A-4) can be written

$$p(|\overline{x} - \mu_x| > \delta) \simeq \lambda = 2\left[1-\Phi(\alpha)\right] \tag{A-6}$$

Substituting (A-3) in (A-5), the number of samples
required can be estimated from

$$N = \left(\frac{\alpha\sigma_x}{\delta}\right)^2 \tag{A-7}$$

Unfortunately, the parameter σ_x^2 is not usually known;
it too must be estimated in the course of the simulation.
An unbiased estimate of σ_x is the standard deviation, s_x,
defined by

$$s_x^2 = \frac{1}{N-1}\left[\sum_{i=1}^{N} x\{j\}^2 - N\overline{x}^2\right], \quad N > 1 \tag{A-8}$$

Then, given an initial estimate of s_x from a few, say 30, samples, the number of samples required can be estimated from

$$N \simeq \left[\frac{\alpha s_x}{\delta}\right]^2 \qquad \text{(A-9)}$$

where s_x has been substituted for σ_x in (A-7).

A more convenient procedure would be to continue sampling until the precision requirement is satisfied, that is

$$\frac{\alpha s_x}{N^{\frac{1}{2}}} \leq \delta \qquad \text{(A-10)}$$

Given λ , the confidence level, α is defined by (A-6).

Shreider suggest that $\lambda = 0.95$, a 95% level of confidence, is suitable which corresponds to $\alpha = 1.96$

2. Calculation of $p(|\beta - \mu| > \delta)$

This calculation is based on the definition and table of the standardized cumulative normal distribution function, $\Phi(t)$, found in Hadley's text (30).

$$p(|\beta_i - \mu_{\beta_i}| > \delta_{\beta_i}) = \Phi\left(\frac{-\delta_{\beta_i}}{\sigma_{\beta_i}}\right) + 1 - \Phi\left(\frac{\delta_{\beta_i}}{\sigma_{\beta_i}}\right)$$

$$= 2\left[1 - \Phi\left(\frac{\delta_{\beta_i}}{\sigma_{\beta_i}}\right)\right] \qquad \text{(A-11)}$$

Table A-1 details the calculation for each β_i.

The probability that a critical parameter, β_i, would violate an upper or lower bound, is, in the worst case (β_1, β_4, β_5, β_6), equivalent to

$$p(\beta_5 > 1.0) = p(\beta_5 - 0.9 > 0.1)$$

$$= 1 - \Phi\left[\frac{0.1}{\sigma_{\beta_5}}\right] = 1 - \Phi(5) \qquad \text{(A-12)}$$

From the table of $\Phi(t)$,

$$\Phi(t) = 1.0000 \quad , \quad t > 3.9 \tag{A-13}$$

Hence, $p(\beta_5 > 1.0) = 0.0000$

TABLE A-1

Calculation of $p(|\beta_i - \mu_{\beta_i}| < \delta_{\beta_i})$

i	σ_{β_i}	δ_{β_i}	$\dfrac{\delta_{\beta_i}}{\sigma_{\beta_i}}$	$\Phi\left(\dfrac{\delta_{\beta_i}}{\sigma_{\beta_i}}\right)$	$(\lvert\beta_i - \mu_{\beta_i}\rvert < \delta_{\beta_i})$
1	0.02	0.05	2.5	0.9938	0.0124
2	0.02	0.05	2.5	0.9938	0.0124
3	0.02	0.05	2.5	0.9938	0.0124
4	0.01	0.25	2.5	0.9938	0.0124
5	0.02	0.05	2.5	0.9938	0.0124
6	0.02	0.05	2.5	0.9938	0.0124

APPENDIX B

VERIFICATION OF MODEL USING MPS/360

Table of Contents

LIST OF TABLES

```
      C     MAINLINE -- E.P.

C     MAINLINE -- E.P.   EXPANDED PROBLEM

C     THIS PROGRAM IS STEP 1 OF A TWO STEP PROCESS
C     OPTIMIZATION PROCEDURE.  IN THIS STEP, PROCESS DATA
C     ARE READ IN AND TABLEAU ENTRIES GENERATED.  L.P. IDENT
C     -IFICATION DATA ARE READ IN AND THE L.P. PROBLEM IS
C     WRITTEN OUT (ON LOGICAL UNIT 7) IN A FORM SUITABLE FOR
C     INPUT TO IBM'S MPS/360 MATHEMATICAL PROGRAMMING
C     PACKAGE.
C
C
C                         - DEFINITIONS -
C
C     FX(K)          -KTH EXTERNAL FRESH FEED
C
C     G(J,K)         -TOTAL FEED OF COMPONENT J TO UNIT K
C
C     P(J,K)         -COMPONENT J OF THE KTH EXTERNAL PRODUCT
C
C     X(J)           -THE VARIABLES OF THE L.P. PROBLEM
C
C
C                        - VARIABLE LIST -
C
C     INPUT VARIABLES -
C
C     A(I,J,K)       -FRACTION OF G(I,J) WHICH GOES TO UNIT K
C                    -NAGIEV OR RECOVERY FACTOR
C
C     CINS(I,J,K)    -COEFFICIENT OF G(J,K) IN THE ITH CON-
C                     STRAINT
C
C     CON(I)         -FRACTION OF G(1,3) WHICH WOULD BE CON-
C                     VERTED TO ( OR DISAPPEAR FROM) COMPONENT
C                     I IN UNIT 3 IF THE SELECTIVITY OF THE
C                     CATALYST WERE 1.0
C
C     CPROD(I,J,K)   -COEFFICIENT OF P(J,K) IN THE ITH CON-
C                     STRAINT
C
C     CREDVA(I,K)    -COEFFICIENT OF FX(K) IN THE ITH CON-
C                     STRAINT
C
C     D(I,J,K)       -FRACTION OF G(I,K) WHICH LEAVES THE
C                     PROCESS AS EXTERNAL PRODUCT P(I,J)
C
C     FBND(I,1)      -BOUND ON FX(I)
C     FBND(I,2)      -FLAG INDICATING TYPE OF BOUND
C                    -SAME DEFINITION AS NBT
C
```

```
      C     MAINLINE -- E.P.  ...(CONT'D)
C     FDCOST(I,J)  -COST OF COMPONENT I OF FX(J)
C
C     GBND(I,K,1)  -BOUND ON G(I,K)
C     GBND(I,K,2)  -FLAG INDICATING TYPE OF BOUND
C                  -SAME AS NBT
C
C     NCODE(I)     -FLAG INDICATING TYPE OF ITH CONSTRAINT
C                       -1 - GREATER THAN OR EQUAL TO
C                        0 - EQUALITY
C                       +1 - LESS THAN OR EQUAL TO
C
C     NCOMP        -NUMBER OF COMPONENTS
C
C     NCON         -NUMBER OF CONSTRAINTS (EXCEPT MATERIAL
C                   BALANCE CONSTRAINTS)
C
C     NG           -NUMBER OF UNITS - INCLUDING STREAM
C                   SPLITTERS
C
C     NF           -NUMBER OF EXTERNAL FRESH FEEDS
C
C     NPRO         -NUMBER OF EXTERNAL PRODUCT STREAMS
C
C     PBND(I,J,1)  -BOUND ON P(I,J)
C     PBND(I,J,2)  -TYPE OF BOUND ON P(I,J)
C                  -SAME AS NBT
C
C     RNGE(I)      -OTHER LIMIT ON RANGE OF RHS(I)
C
C     RHS(I)       -RIGHT HAND SIDE OF CONSTRAINT I
C
C     SEL          -SELECTIVITY OF CATALYST IN UNIT 3
C
C     VALPRO(I,J)  -VALUE OF PRODUCT STREAM P(I,J)
C
C     Y(I,J,K)     -FRACTION OF FEED FX(K), FED TO UNIT K,
C                   WHICH IS COMPONENT I
C     TABLEAU VARIABLES -
C
C     BOUND(K)     -THE BOUND ON THE KTH VARIABLE (COLUMN)
C                   IN THE LINEAR PROGRAMMING PROBLEM
C
C     CC(J,K)      -THE COEFFICIENT IN THE JTH ROW, KTH
C                   COLUMN IN THE COEFFICIENT MATRIX OF THE
C                   SET OF CONSTRAINTS
C
C     CZ(K)        -THE COST OF THE KTH VARIABLE
C                  -THE OBJECTIVE FUNCTION
C
C     M            -THE NUMBER OF ROWS IN THE L.P.PROBLEM
```

```
      C      MAINLINE -- E.P.  ...(CONT'D)

C                        EXCLUDING THE OBJECTIVE FUNCTION ROW
C                       -THE NUMBER OF CONSTRAINTS EXCLUDING
C                        RANGES AND BOUNDS
C
C     N                 -THE NUMBER OF COLUMNS IN THE L.P. PRO-
C                        BLEM EXCLUDING LOGICAL VARIABLES AND THE
C                        RIGHT HAND SIDE
C
C     R(J)              -THE RIGHT HAND SIDE OF THE JTH CON-
C                        STRAINT
C
C     RANGE(K)          -IF R(K) MAY VARY OVER A RANGE OF VALUES,
C                        R(K) IS ONE LIMIT ON THE RANGE (UPPER OR
C                        LOWER) AND RANGE(K) IS THE MAGNITUDE BY
C                        WHICH THE RIGHT-HAND SIDE MAY VARY FROM
C                       THE LIMIT PREVIOUSLY SPECIFIED
C
C
C     CONTROL VARIABLES -
C
C     MM                -THE NUMBER OF ROWS CC IS DIMENSIONED FOR
C                       -USED FOR ADJUSTABLE DIMENSIONS
C
C     NB                -INPUT FLAG INDICATING THE PRESENCE OF A
C                        BOUND VECTOR
C                            NB=0 - NO BOUNDS
C                            NB=1 - BOUNDS REQUIRED
C
C     NBT(K)            -FLAG INDICATING TYPE OF BOUND
C                            0 - NO BOUND ON THIS COLUMN
C                            1 - LOWER BOUND
C                            2 - UPPER BOUND
C                            3 - FIXED VALUE
C                            4 - FREE VARIABLE
C                            5 - LOWER BOUND IS -INFINITY
C                            6 - UPPER BOUND IS +INFINITY
C
C     NN                -NUMBER OF COLUMNS CC IS DIMENSIONED FOR
C                       -USED FOR ADJUSTABLE DIMENSIONING
C
C     NR                -INPUT FLAG INDICATING PRESENCE OF RANGE
C                        VECTOR
C                            NR=0 - NO RANGE VECTOR
C                            NR=1 - RANGE VECTOR REQUIRED
C
C     NSIG(K)           -FLAG INDICATING THE TYPE OF THE KTH
C                        CONSTRAINT IN THE TABLEAU
C                            -1 - GREATER THAN OR EQUAL TO
C                             0 - EQUALITY
C                            +1 - LESS THAN OR EQUAL TO
```

```
      C     MAINLINE -- E.P.  ...(CONT'D)
C
C
C     THIS PROGRAM GENERATES THE GENERAL (EXPANDED) FORM OF
C     THE L.P. PROBLEM.
C
C
C  PROCESS DATA SPECIFICATION

      COMMON A(4,15,15),Y(4,15,11),D(4,6,15),CPROD(10,4,6),
     1CINS(10,4,15),CREDVA(10,11),FDCOST(4,11),
     1OPCOST(4,15),VALPRO(4,6),CON(4),RHS(10),RNGE(10),
     1GBND(4,15,2),PBND(4,6,2),FBND(11,2),SEL,NCODE(10),
     1NCOMP,NPRO,NG,NF,NCON

C  L.P. DATA SPECIFICATION

      REAL CC(100,100),BOUND(100),R(100),CZ(100),RANGE(100)
      INTEGER NBT(100),NSIG(100)
      MM=100
      NN=100

C  INPUT PROCESS DATA

      CALL INPUT(NB,NR)

C  GENERATE L.P. TABLEAU ENTRIES

      CALL SFEBDP(CC,R,CZ,RANGE,BOUND,NBT,NSIG,NB,NR,M,N,
     1MM,NN)

C  READ L.P. IDENTIFICATION AND WRITE GENERATED L.P. PROBLEM
C  IN MPS DATA FORMAT

      CALL MPSDAT(CC,R,CZ,RANGE,BOUND,NBT,NSIG,NB,NR,M,N,
     1MM,NN)
      STOP
      END
```

C SUBROUTINE SFEBDP

```
C     SUBROUTINE SFEBDP
C
C      SUBROUTINE SFEBDP GENERATES THE L.P. COEFFICIENTS FOR
C      THE EXPANDED FORM OF THE L.P. PROBLEM.  ALL VARIABLES
C      IN THE ARGUMENT LIST AND IN COMMON ARE DEFINED IN THE
C      MAINLINE.
C
C
C      FOR THE EXPANDED PROBLEM
C           (X(J),J=1,N) = (FX(J),J=1,NF),
C                          ((G(K,L),K=1,NCOMP),L=1,NG),
C                          ((P(K,L),K=1,NCOMP),L=1,NPRO)
C

      SUBROUTINE SFEBDP(CC,R,CZ,RANGE,BOUND,NBT,NSIG,
     1NB,NR,M,N,MM,NN)


      COMMON A(4,15,15),Y(4,15,11),D(4,6,15),CPROD(10,4,6),
     1CINS(10,4,15),CREDVA(10,11),FDCOST(4,11),
     1OPCOST(4,15),VALPRO(4,6),CON(4),RHS(10),RNGE(10),
     1GBND(4,15,2),PBND(4,6,2),FBND(11,2),SEL,NCODE(10),
     1NCOMP,NPRO,NG,NF,NCON
      DIMENSION CC(MM,NN),BOUND(NN),NBT(NN),R(MM),CZ(NN),
     1RANGE(MM),NSIG(MM)


  500 FORMAT('1',//////13X,'SFEBDP - THE GENERATION OF THE',
     1' TABLEAU FOR THE EXPANDED '/16X,'FORM OF THE ',
     2'L.P. PROBLEM.  THE MATERIAL BALANCE EQUATIONS ',
     4/1H ,15X,'ARE INCLUDED EXPLICITLY.'/)
  501 FORMAT(1H0,12X,'THE INITIALIZATION OF THE TABLEAU ',
     1'TO ZERO IS COMPLETE.'/1H ,15X,'THE PROBLEM WILL ',
     2'HAVE',I3,' ROWS (CONSTRAINTS) AND'/1H ,15X,I3,
     3' COLUMNS (VARIABLES).')
  502 FORMAT(1H0,12X,'THE OBJECTIVE FUNCTION HAS BEEN ',
     1'ENTERED')
  503 FORMAT(1H0,12X,'THE INEQUALITY CONSTRAINTS HAVE ',
     1'BEEN ENTERED')
  504 FORMAT(1H0,12X,'THE MATERIAL BALANCE EQUATIONS ON ',
     1'THE INTERNAL STREAMS'/1H ,15X,'HAVE BEEN ENTERED')
  505 FORMAT(1H0,12X,'THE MATERIAL BALANCE EQUATIONS ON ',
     1'THE PRODUCT STREAMS '/1H ,15X,'HAVE BEEN ENTERED')
  506 FORMAT(1H0,12X,'THE BOUNDS HAVE BEEN ENTERED')
  507 FORMAT(1H0,12X,'SFEBDP FINISHED')


      WRITE(6,500)
```

```
      C     SUBROUTINE SFEBDP  ...(CONT'D)
C     INITIALIZE TABLEAU AND CONTROL VARIABLES.
C

      M=NG*NCOMP+NPRO*NCOMP+NCON
      N=NG*NCOMP+NPRO*NCOMP+NF
      DO 10 J=1,M
      NSIG(J)=0
      R(J)=0.
      RANGE(J)=0.
      DO 20 K=1,N
      CC(J,K)=0.
   20 CONTINUE
   10 CONTINUE
      DO 30 J=1,N
      NBT(J)=0
      BOUND(J)=0.
      CZ(J)=0.
   30 CONTINUE
      WRITE(6,501) M,N

C     THE OBJECTIVE FUNCTION HAS THE FORM
C     MIN Z = SUM OF FEED COSTS + SUM OF OPERATING COSTS
C                 + SUM OF PRODUCT VALUES
C           = SUM((J=1,NCOMP),K=1,NF) FDCOST(J,K)*F(J,K)
C            + SUM((J=1,NCOMP),K=1,NG) OPCOST(J,K)*G(J,K)
C            + SUM((J=1,NCOMP),K=1,NPRO) VALPRO(J,K)*P(J,K)
C           = SUM (J=1,N) CZ(J)*X(J)
C     WHERE F(J,K) = (SUM (L=1,NG) Y(J,L,K))*FX(K)
C
C     CALCULATION OF CZ(J)
C
C     FEED STREAM COEFFICIENTS

      DO 100 J=1,NCOMP
      DO 100 K=1,NF
      SUM=0
      DO 90 L=1,NG
   90 SUM=SUM+Y(J,L,K)
      CZ(K)=CZ(K)+FDCOST(J,K)*SUM
  100 CONTINUE

C     INTERNAL STREAM COEFFICIENTS

      NV=NF
      DO 120 K=1,NG
      DO 120 J=1,NCOMP
      NV=NV+1
      CZ(NV)=OPCOST(J,K)
  120 CONTINUE
```

```
            C     SUBROUTINE SFEBDP  ...(CONT'D)
C     PRODUCT STREAM COEFFICIENTS

      DO 140 K=1,NPRO
      DO 140 J=1,NCOMP
      NV=NV+1
      CZ(NV)=-VALPRO(J,K)
  140 CONTINUE
      WRITE(6,502)
      WRITE(6,502)

C     THE JJTH PROCESS OR IMPLIED CONSTRAINT HAS THE FORM
C     RHS(JJ)  (.LE. , .EQ. , OR .GE.)
C        SUM (K=1,NF) CREDVA(J,K)*FX(K)
C         + SUM ((L=1,NCOMP),K=1,NG) CINS(J,L,K)*G(L,K)
C         + SUM ((L=1,NCOMP),K=1,NPRO) CPROD(J,L,K)*P(L,K)
C                (.GE. , .EQ. , OR .LE.)  RNGE(JJ)
C     OR
C     RHS(JJ) (.LE.,.EQ., OR.GE.) SUM (K=1,N) CC(JJ,K)*X(K)
C           (.GE., .EQ., OR .LE.) RANGE(JJ)
C     WHERE
C       JJ=1,NCON
C
C     CALCULATION OF CC(JJ,N)
C

      NE=NCON
      IF(NCON) 235,235,195
  195 DO 230 J=1,NCON
      DO 200 K=1,NF
      CC(J,K)=CREDVA(J,K)
  200 CONTINUE
      NV=NF
      DO 210 K=1,NG
      DO 210 L=1,NCOMP
      NV=NV+1
      CC(J,NV)=CINS(J,L,K)
  210 CONTINUE
      NV=NF+NCOMP*NG
      DO 220 K=1,NPRO
      DO 220 L=1,NCOMP
      NV=NV+1
      CC(J,NV)=CPROD(J,L,K)
  220 CONTINUE
  230 CONTINUE

C     CALCULATE THE RHS'S,CONSTRAINT TYPES AND RANGES
C

  235 DO 236 J=1,NCON
      R(J)=RHS(J)
```

```
C    SUBROUTINE SFEBDP  ...(CONT'D)

      NSIG(J)=NCODE(J)
  236 CONTINUE
      IF(NR) 239,239,237
  237 DO 238 J=1,NCON
      RANGE(J)=RNGE(J)
  238 CONTINUE
      WRITE(6,503)

C     MATERIAL BALANCE EQUATIONS
C
C
C     THE JJTH MATERIAL BALANCE EQUATION ON THE INTERNAL
C     STREAMS HAS THE FORM
C     0 = -SUM (L=1,NF) Y(K,J,L)*FX(L)
C         +G(K,J) -SUM (L=1,NG) A(K,L,J)*G(K,L)
C       = SUM (L=1,N) CC(JJ,L)*X(L)
C     WHERE
C       (JJ=NCON,(NCON+NG*NCOMP)) - ((K=1,NCOMP),J=1,NG)
C
C     CALCULATION OF CC(JJ,L)
C

  239 DO 250 J=1,NG
      DO 250 K=1,NCOMP
      NE=NE+1

C     FEED STREAM COEFFICIENTS

      DO 240 L=1,NF
      IF(ABS(Y(K,J,L)).LT.0.000001) GOTO 240
      CC(NE,L)=-Y(K,J,L)
  240 CONTINUE


C     INTERNAL STREAM COEFFICIENTS

      DO 245 L=1,NG
      NV=NF+NCOMP*(L-1)+K
      IF(ABS(A(K,L,J)).LT.0.000001) GOTO 242
      CC(NE,NV)=-A(K,L,J)
  242 IF(L-J) 245,243,245
  243 CC(NE,NV)=CC(NE,NV)+1.
  245 CONTINUE

C     IF THE PROCESS CONTAINS A REACTION VESSAL, THE
C     ADDITION OF A COMPONENT MUST BE PROVIDED FOR BY THE
C     USE OF A PHANTOM FEED STREAM.
C
C     FOR THIS PLANT, REACTION OCCURS IN UNIT 3 SO A PHANTOM
C     FEED IS ADDED AT UNIT 3.  THE AMOUNT OF COMPONENT K
```

```
          C     SUBROUTINE SFEBDP  ...(CONT'D)
C      ADDED  = CON(K)*SEL*G(1,3)
C
       IF(J-4) 250,246,250
  246  NV=NF+2*NCOMP+1
       CC(NE,NV)=CC(NE,NV)-A(K,3,J)*SEL*CON(K)
  250  CONTINUE
       WRITE(6,504)
C
C      THE JJTH MATERIAL BALANCE EQUATION ON THE PRODUCT
C      STREAMS HAS THE FORM
C      0 = P(K,J) - SUM (L=1,NG) D(K,J,L)*G(K,L)
C        = SUM (L=1,N) CC(JJ,L)*X(L)
C      WHERE
C        (JJ=(NCON+NG*NCOMP),M) - ((K=1,NCOMP),J=1,NPRO)
C
C      CALCULATION OF CC(JJ,L)
C
       DO 280 J=1,NPRO
       DO 280 K=1,NCOMP
       NE=NE+1
C
C      INTERNAL STREAM COEFFICIENTS
C
       DO 260 L=1,NG
       NV=NF+NCOMP*(L-1)+K
       IF(ABS(D(K,J,L)).LT.0.00001) GOTO 260
       CC(NE,NV)=-D(K,J,L)
  260  CONTINUE
C
C      PRODUCT STREAM COEFFICIENTS
C
       NV=NF+NCOMP*NG+NCOMP*(J-1)+K
       CC(NE,NV)=1.
  280  CONTINUE
       WRITE(6,505)
C
C      THE BOUND VECTOR LISTS THE BOUNDS ON THE VARIABLES
C      X(J).  THE CONTROL VECTOR NBT INDICATES THE TYPE OF
C      BOUND  ON X(J).  IT'S FORM IS EXPLAINED IN THE
C      MAINLINE DEFINITION.
C
C      CALCULATION OF BOUNDS, NBT VECTORS.
C
  289  IF(NB) 321,321,290
C
C      FEED STREAM BOUNDS
```

```
      C    SUBROUTINE SFEBDP   ...(CONT'D)

  290 DO 300 J=1,NF
      BOUND(J)=FBND(J,1)
      NBT(J)=FBND(J,2)
  300 CONTINUE
C     INTERNAL STREAM BOUNDS
      NV=NF
      DO 310 J=1,NG
      DO 310 L=1,NCOMP
      NV=NV+1
      BOUND(NV)=GBND(L,J,1)
      NBT(NV)=GBND(L,J,2)
  310 CONTINUE
C     PRODUCT STREAM BOUNDS
      DO 320 J=1,NPRO
      DO 320 L=1,NCOMP
      NV=NV+1
      BOUND(NV)=PBND(L,J,1)
      NBT(NV)=PBND(L,J,2)
  320 CONTINUE
      WRITE(6,506)
  321 CONTINUE
      WRITE(6,507)
      RETURN
      END
```

C SUBROUTINE MPSDAT

```
C     SUBROUTINE MPSDAT

C
C      SUBROUTINE MPSDAT READS IN PARAMETER IDENTIFICATION
C      INFORMATION (ROW NAMES ETC) AND WRITES THE L.P.
C      PROBLEM IN A FORM SUITABLE FOR INPUT TO MPS/360
C
C      THE FUNCTION LINCT IS USED TO CONTROL THE LINE COUNT
C      ON THE PRINTED OUTPUT.
C

      SUBROUTINE MPSDAT(CON,RH,CZ,RA,BN,NBT,SIG,NB,NR,
     1M,N,MM,NN)


      REAL CON(MM,NN),RH(MM),CZ(NN),RA(MM),BN(NN)
      INTEGER SIG(MM),NBT(NN)
      INTEGER NMP(5,8)
      REAL RT(4)/'N','G','E','L'/,BT(6)/'LO','UP','FX','FR',
     1'MI','PL'/
      DIMENSION RNME(120,2),CNME(200,2),RANM(2),BNNM(2),
     1OBJNME(2),RHSNME(2),DSNME(2)
      DATA ROWS,COLU,MNS,RHS,RANG,ES,BOUN,DS,ENDA,TA,NAME/
     1'ROWS','COLU','MNS','RHS','RANG','ES','BOUN','DS',
     2'ENDA','TA','NAME'/


  605 FORMAT(21X,A2,1X,2A4,2X,2A4,2X,F12.6)
  105 FORMAT(1X,A2,1X,2A4,2X,2A4,2X,F12.6)
  604 FORMAT(21X,A2,1X,2A4,2X,2A4,2X,G16.6)
  104 FORMAT(1X,A2,1X,2A4,2X,2A4,2X,G16.6)
  603 FORMAT(24X,2A4,2X,2A4,2X,F12.6)
  103 FORMAT(4X,2A4,2X,2A4,2X,F12.6)
  607 FORMAT(24X,A4,T35,2A4,T60,2A4)
  107 FORMAT(4X,A4,T15,2A4,T40,2A4)
  602 FORMAT(24X,2A4,2X,2A4,2X,G16.6)
  102 FORMAT(4X,2A4,2X,2A4,2X,G16.6)
  601 FORMAT(21X,A1,2X,2A4)
  101 FORMAT(1X,A1,2X,2A4)
  600 FORMAT(20X,2A4)
  106 FORMAT(A4,T15,2A4)
  606 FORMAT(20X,A4,T35,2A4)
  100 FORMAT(2A4)
   10 FORMAT(2X,2A4)
  500 FORMAT(5A4)


      NLT=LINCT(52,0)
      NLT=LINCT(NLT,3)
```

```
          C    SUBROUTINE MPSDAT  ...(CONT'D)
C      READ AND PUNCH DATA SET NAME
C

       READ(5,10)(DSNME(K),K=1,2)
       WRITE(7,106) NAME,(DSNME(K),K=1,2)
       WRITE(6,606) NAME,(DSNME(K),K=1,2)

C      READ ROW NAMES, COLUMN NAMES AND OBJECTIVE FUNCTION
C      NAME.
C

       READ(5,10)(RNME(J,1),RNME(J,2),J=1,M)
       READ(5,10)(CNME(J,1),CNME(J,2),J=1,N)
       READ(5,10) OBJNME(1),OBJNME(2)

C      PUNCH ROWS SECTION
C

       WRITE(7,100) ROWS
       WRITE(6,600) ROWS
       WRITE(7,101) RT(1),(OBJNME(J),J=1,2)
       WRITE(6,601) RT(1),(OBJNME(J),J=1,2)
       DO 200 J=1,M
       NLT=LINCT(NLT,1)
       WRITE(7,101) RT(3+SIG(J)),(RNME(J,K),K=1,2)
       WRITE(6,601) RT(3+SIG(J)),(RNME(J,K),K=1,2)
   200 CONTINUE

C      PUNCH COLUMNS SECTION.


       WRITE(7,100) COLU,MNS
       NLT=LINCT(NLT,1)
       WRITE(6,600) COLU,MNS
       DO 210 J=1,N
       IF(ABS(CZ(J)).LT.1.0E-06) GOTO 202
       NLT=LINCT(NLT,1)
       IF(ABS(CZ(J)).LT.0.1000001) GOTO 201
       WRITE(7,102)(CNME(J,K),K=1,2),(OBJNME(K),K=1,2),CZ(J)
       WRITE(6,602)(CNME(J,K),K=1,2),(OBJNME(K),K=1,2),CZ(J)
       GOTO 202
   201 WRITE(7,103)(CNME(J,K),K=1,2),(OBJNME(K),K=1,2),CZ(J)
       WRITE(6,603)(CNME(J,K),K=1,2),(OBJNME(K),K=1,2),CZ(J)
   202   DO 205 K=1,M
       IF(ABS(CON(K,J)).LT.1.0E-06) GOTO 205
       NLT=LINCT(NLT,1)
       IF(ABS(CON(K,J)).LT.0.1000001) GOTO 206
       WRITE(7,102)(CNME(J,L),L=1,2),(RNME(K,L),L=1,2),
      1CON(K,J)
       WRITE(6,602)(CNME(J,L),L=1,2),(RNME(K,L),L=1,2),
```

```
      C    SUBROUTINE MPSDAT  ...(CONT'D)

     1CON(K,J)
      GOTO 205
  206 WRITE(7,103)(CNME(J,L),L=1,2),(RNME(K,L),L=1,2),
     1CON(K,J)
      WRITE(6,603)(CNME(J,L),L=1,2),(RNME(K,L),L=1,2),
     1CON(K,J)
  205 CONTINUE
  210 CONTINUE
C     PUNCH RHS SECTION.
C
  215 WRITE(7,100) RHS
      NLT=LINCT(NLT,1)
      WRITE(6,600) RHS
      READ(5,10) RHSNME(1),RHSNME(2)
      DO 240 J=1,M
      IF(J.EQ.1) GOTO 220
      IF(ABS(RH(J)).LT.1.0E-06) GOTO 240
  220 NLT=LINCT(NLT,1)
      IF(ABS(RH(J)).LT.0.1000001) GOTO 230
      WRITE(7,102) (RHSNME(K),K=1,2),(RNME(J,K),K=1,2),
     1RH(J)
      WRITE(6,602) (RHSNME(K),K=1,2),(RNME(J,K),K=1,2),
     1RH(J)
      GOTO 240
  230 WRITE(7,103) (RHSNME(K),K=1,2),(RNME(J,K),K=1,2),
     1RH(J)
      WRITE(6,603) (RHSNME(K),K=1,2),(RNME(J,K),K=1,2),
     1RH(J)
  240 CONTINUE
C     PUNCH RANGES SECTION.
C
      IF(NR) 300,300,250
  250 WRITE(7,100) RANG,ES
      NLT=LINCT(NLT,1)
      WRITE(6,600) RANG,ES
      READ(5,10) RANM(1),RANM(2)
      DO 255 K=1,M
      IF(K.EQ.1) GOTO 251
      IF(ABS(RA(K)).LT.1.0E-06) GOTO 255
  251 NLT=LINCT(NLT,1)
      IF(ABS(RA(K)).LT.0.1000001) GOTO 252
      WRITE(7,102) (RANM(L),L=1,2),(RNME(K,L),L=1,2),RA(K)
      WRITE(6,602) (RANM(L),L=1,2),(RNME(K,L),L=1,2),RA(K)
      GOTO 255
  252 WRITE(7,103) (RANM(L),L=1,2),(RNME(K,L),L=1,2),RA(K)
      WRITE(6,603) (RANM(L),L=1,2),(RNME(K,L),L=1,2),RA(K)
```

```
      C     SUBROUTINE MPSDAT   ...(CONT'D)

  255 CONTINUE

C     PUNCH BOUNDS SECTION.
C

  300 IF(NB) 350,350,310
  310 WRITE(7,100) BOUN,DS
      NLT=LINCT(NLT,1)
      WRITE(6,600) BOUN,DS
      READ(5,10)(BNNM(K),K=1,2)
      DO 330 K=1,N
      IF(K.EQ.1) GOTO 315
      IF(NBT(K).LE.0) GOTO 330
  315 NLT=LINCT(NLT,1)
      IF(ABS(BN(K)).LT.0.1000001) GOTO 320
      WRITE(7,104)BT(NBT(K)),(BNNM(L),L=1,2),(CNME(K,L),
     1L=1,2),BN(K)
      WRITE(6,604)BT(NBT(K)),(BNNM(L),L=1,2),(CNME(K,L),
     1L=1,2),BN(K)
      GOTO 330
  320 WRITE(7,105)BT(NBT(K)),(BNNM(L),L=1,2),(CNME(K,L),
     1L=1,2),BN(K)
      WRITE(6,605)BT(NBT(K)),(BNNM(L),L=1,2),(CNME(K,L),
     1L=1,2),BN(K)
  330 CONTINUE
  350 WRITE(7,100) ENDA,TA
      NLT=LINCT(NLT,1)
      WRITE(6,600) ENDA,TA

C     IF FILE 7 IS TO BE STORED FOR LATER INPUT TO MPS
C     RATHER THAN ROUTED TO SYSPUNCH, INCLUDE THE FOLLOWING
C     TWO STATEMENTS.
C
C     END FILE 7
C     REWIND 7
C

      RETURN
      END
```

C FUNCTION LINCT

```
C     FUNCTION LINCT
C
C     THE FUNCTION LINCT IS USED TO CONTROL THE NUMBER OF
C     LINES PRINTED.
C
      FUNCTION LINCT(NLT,NL)
      LINCT=NLT+NL
      IF(LINCT-51) 15,15,10
   10 WRITE(6,11)
   11 FORMAT('1',/////)
      LINCT=0
   15 RETURN
      END
```

C MAINLINE -- R.P.

```
C     MAINLINE -- R.P.   REDUCED PROBLEM

C     THIS PROGRAM GENERATES THE REDUCED FORM OF THE L.P.
C     PROBLEM
C
C     VARIABLES ARE AS DEFINED FOR MAINLINE -- E.P.
C
C
C     PROCESS DATA SPECIFICATION

      COMMON A(4,15,15),Y(4,15,11),D(4,6,15),CPROD(10,4,6),
     1CINS(10,4,15),CREDVA(10,11),FDCOST(4,11),
     1OPCOST(4,15),VALPRO(4,6),CON(4),RHS(10),RNGE(10),
     1GBND(4,15,2),PBND(4,6,2),FBND(11,2),SEL,NCODE(10),
     1NCOMP,NPRO,NG,NF,NCON

C     L.P. DATA SPECIFICATION

      DIMENSION CC(15,11),R(15),CZ(11),RANGE(15),BOUND(11),
     1NBT(11),NSIG(15)
      MM=15
      NN=11

C  INPUT PROCESS DATA

      CALL INPUT(NB,NR)

C     INVERT B MATRIX

      CALL BALAN

C     GENERATE L.P. TABLEAU ENTRIES

      CALL SFCBDP(CC,R,CZ,RANGE,BOUND,NBT,NSIG,NB,
     1M,N,MM,NN)

C     READ L.P. IDENTIFICATION AND WRITE GENERATED L.P.
C     PROBLEM IN MPS DATA FORMAT

      CALL MPSDAT(CC,R,CZ,RANGE,BOUND,NBT,NSIG,NB,NR,
     1M,N,MM,NN)
      STOP
      END
```

SUBROUTINE BALAN

```
      SUBROUTINE BALAN
C     SUBROUTINE BALAN INVERTS THE MATERIAL BALANCE MATRIX
C     B.  USING THE INVERSE IT CALCULATES THE MATRICES BIF
C     AND PBIF WHICH ARE USED TO EXPRESS G(J,K) AND P(J,K)
C     AS LINEAR COMBINATIONS OF THE FEED STREAMS FX(K).
C
      COMMON A(4,15,15),Y(4,15,11),D(4,6,15),CPROD(10,4,6),
     1CINS(10,4,15),CREDVA(10,11),FDCOST(4,11),
     1OPCOST(4,15),VALPRO(4,6),CON(4),RHS(10),RNGE(10),
     1GBND(4,15,2),PBND(4,6,2),FBND(11,2),SEL,NCODE(10),
     1NCOMP,NPRO,NG,NF,NCON
      COMMON /BAL/ BIF(60,11),PBIF(24,11)
      REAL B(60,60),WV1(60),WV2(60)
    1 FORMAT('1',//////,13X,'BALAN - USES THE MATERIAL ',
     1'BALANCE EQUATIONS TO SOLVE FOR'/16X,'THE INTERNAL ',
     3'AND PRODUCT STREAMS AS FUNCTIONS OF THE ',/1H ,15X,
     4'FRESH FEED STREAMS.  THESE FUNCTIONS ARE REQUIRED ',
     5'LATER ',/1H ,15X, 'BY SFCBDP',/)
    2 FORMAT(1H0,12X,'THE MATERIAL BALANCE MATRIX B HAS ',
     1'BEEN SET UP')
    3 FORMAT(1H0,12X,'B HAS BEEN INVERTED')
    4 FORMAT(1H0,12X,'THE MATRIX BIF DEFINING THE INTERNAL',
     1' STREAMS AS FUNCTIONS '/1H ,15X,'OF THE FRESH FEEDS',
     2' HAS BEEN GENERATED')
    5 FORMAT(1H0,12X,'THE MATRIX PBIF DEFINING THE PRODUCT',
     1' STREAMS AS FUNCTIONS '/1H ,15X,'OF THE FRESH FEEDS',
     2' HAS BEEN GENERATED')
    6 FORMAT(1H0,12X,'BALAN FINISHED')
      WRITE(6,1)
C     THE MATERIAL BALANCE MATRIX B HAS THE FORM   (I - AT)
C     WHERE   I IS AN IDENTITY MATRIX OF ORDER NG*NCOMP
C             AT IS THE TRANSPOSE OF THE NAGIEV RECOVERY
C                FACTOR MATRIX.  A(J,L,K) IS THE ELEMENT IN
C                THE IITH ROW AND JJTH COLUMN OF AT
C             II = (K-1)*NCOMP+J
C             JJ = (L-1)*NCOMP+J
C
C     CALCULATION OF B
C
      DO 35 J=1,60
      DO 35 K=1,60
      B(J,K)=0.
   35 CONTINUE
      NE=0
      DO 50 K=1,NG
      DO 50 J=1,NCOMP
```

SUBROUTINE BALAN ...(CONT'D)

```
      NE=NE+1
      DO 45 L=1,NG
      NV=(L-1)*NCOMP+J
      B(NE,NV)=-A(J,L,K)
      IF(K-L) 45,40,45
   40 B(NE,NV)=B(NE,NV)+1.
   45 CONTINUE
   50 CONTINUE

C     PRODUCTION VIA REACTION IN UNIT 3 IS ACCOUNTED FOR
C     BY THE ADDITION OF A PHANTOM FEED TO UNIT 3
C         = CON(J)*SEL*G(1,3)
C     FOR THE JTH COMPONENT
C

      DO 54 J=1,NCOMP
      NE=NCOMP*(3)+J
      NV=NCOMP*(2)+1
      B(NE,NV)=B(NE,NV)-A(J,3,4)*CON(J)
   54 CONTINUE
      WRITE(6,2)

C     THE MATRIX B IS INVERTED USING THE SUBROUTINE MINV
C     FROM IBM'S SCIENTIFIC SUBROUTINE PACKAGE.  THE ROUTINE
C     RETURNS THE INVERSE AS B.
C

      I60=60
      CALL MINV(B,I60,DET,WV1,WV2)
      WRITE(6,3)
      IF(DET) 60,55,60
   55 WRITE(6,655)
  655 FORMAT(1H0,12X,'NO INVERSE')
      RETURN

C     THE MATERIAL BALANCE ON INTERNAL STREAMS HAS THE FORM
C      G(L,K) = SUM(J=1,NF) (SUM ((LL=1,NCOMP),KK=1,NG)
C                     B(NE,NV)*Y(LL,KK,J) )*FX(J)
C               = SUM (J=1,NF) BIF(NE,J)*FX(J)
C     WHERE
C         (NE=1,NG*NCOMP) = ((L=1,NCOMP),K=1,NG)
C         (NV=1,NG*NCOMP) = ((LL=1,NCOMP),KK=1, NG)
C
C     CALCULATION OF BIF
C

   60 DO 68 J=1,NF
      NE=0
      DO 65 K=1,NG
      DO 65 L=1,NCOMP
```

SUBROUTINE BALAN ...(CONT'D)

```
      NE=NE+1
      NV=0
      BIF(NE,J)=0.
      DO 65 KK=1,NG
      DO 65 LL=1,NCOMP
      NV=NV+1
      BIF(NE,J)=BIF(NE,J)+B(NE,NV)*Y(LL,KK,J)
   65 CONTINUE
   68 CONTINUE
      WRITE(6,4)
C
C     THE MATERIAL BALANCE ON PRODUCT STREAMS HAS THE FORM
C      P(L,K) = SUM(J=1,NF) ( SUM(KK=1,NG) D(L,K,KK)*
C                   BIF(NV,J) )*FX(J)
C             = SUM(J=1,NF) PBIF(NE,J)*FX(K)
C
C     WHERE
C         (NE=1,NG*NCOMP) = ((L=18NCOMP),K=1,NG)
C         (NV=1,NG*NCOMP) = ((LL=1,NCOMP),KK=1,NG)
C
C     CALCULATION OF PBIF
C

      DO 80 J=1,NF
      DO 62 K=1,24
   62 PBIF(K,J)=0.
      NE=0
      DO 70 K=1,NPRO
      DO 70 L=1,NCOMP
      NE=NE+1
      NV=L-NCOMP
      DO 70 KK=1,NG
      NV=NV+NCOMP
      PBIF(NE,J)=PBIF(NE,J)+D(L,K,KK)*BIF(NV,J)
   70 CONTINUE
   80 CONTINUE
      WRITE(6,5)
      WRITE(6,6)
      RETURN
      END
```

C   SUBROUTINE SFCBDP

```
C     SUBROUTINE SFCBDP

C      SUBROUTINE SFCBDP GENERATES THE L.P. COEFFICIENTS FOR
C      THE CONTRACTED FORM OF THE L.P. PROBLEM.  ALL VARI-
C      ABLES IN THE ARGUMENT LIST AND IN COMMON ARE DEFINED
C      IN THE MAINLINE.
C
C      FOR THE CONTRACTED PROBLEM
C         X(J)=FX(J)  (J=1,NF)
C         G(J,K) = SUM(L=1,NF) BIF(NE,L)*FX(L)
C         P(J,K) = SUM(L=1,NF) PBIF(NE,L)*FX(L)
C      WHERE   (NE=1,NG*NCOMP)=((J=1,NCOMP),K=1,NG)
C
C      THE MATRICES BIF AND PBIF HAVE BEEN GENERATED BY BALAN
C

       SUBROUTINE SFCBDP(CC,R,CZ,RANGE,BOUND,NBT,NSIG,
      1NB,M,N,MM,NN)


       COMMON A(4,15,15),Y(4,15,11),D(4,6,15),CPROD(10,4,6),
      1CINS(10,4,15),CREDVA(10,11),FDCOST(4,11),
      1OPCOST(4,15),VALPRO(4,6),CON(4),RHS(10),RNGE(10),
      1GBND(4,15,2),PBND(4,6,2),FBND(11,2),SEL,NCODE(10),
      1NCOMP,NPRO,NG,NF,NCON
       COMMON /BAL/ BIF(60,11),PBIF(24,11)
       DIMENSION CC(MM,NN),R(MM),CZ(NN),RANGE(MM),NSIG(MM),
      1BOUND(NN),NBT(NN)


     1 FORMAT('1',//////,13X,'SFCBDP - THE GENERATION OF ',
      1' THE L.P. PROBLEM.  THE INTERNAL'/16X,'AND PRODUCT ',
      3'STREAMS ARE ELIMINATED BY SUBSTITUTION OF THE ',
      4/16X,'FUNCTIONS OF FRESH FEEDS GENERATED BY BALAN.')
     2 FORMAT(1H0,12X,'THE OBJECTIVE FUNCTION HAS BEEN ',
      1'GENERATED AS A FUNCTION '/1H ,15X,'OF FRESH FEEDS ',
      3'ONLY')
     3 FORMAT(1H0,12X,'THE INEQUALITY CONSTRAINTS HAVE ',
      2'BEEN GENERATED AS FUNCTIONS ',/1H ,15X,'OF FRESH ',
      2'FEEDS ONLY')
     4 FORMAT(1H0,12X,'THE BOUNDS ON THE FRESH FEEDS HAVE',
      1' BEEN ENTERED.  BOUNDS '/1H ,15X,'ON INTERNAL AND',
      2' PRODUCT STREAMS HAVE BEEN '/1H ,15X,'CONVERTED TO ',
      3'INEQUALITY CONSTRAINTS')
     5 FORMAT(1H0,12X,'SFCBDP FINISHED')


       WRITE(6,1)

C      THE OBJECTIVE FUNCTION HAS THE FORM
```

```
      C     SUBROUTINE SFCBDP  ...(CONT'D)
C     MIN Z = SUM((J=1,NCOMP),K=1,NF) FDCOST(J,K)*F(J,K)
C             + SUM((J=1,NCOMP),K=1,NG) OPCOST(J,K)*G(J,K)
C             + SUM((J=1,NCOMP),K=1,NPRO) VALPRO(J,K)*P(J,K)
C           = SUM (J=1,NF) CZ(J)*X(J)
C     WHERE F(J,K) = (SUM (L=1,NG) Y(J,L,K))*FX(K)
C
C     CALCULATION OF CZ(J)
C

      M=0
      N=NF
      DO 100 K=1,NF
      CZ(K)=0.
      DO 100 J=1,NCOMP
      SUM=0
      DO 95 L=1,NG
   95 SUM=SUM+Y(J,L,K)
      CZ(K)=CZ(K)+FDCOST(J,K)*SUM
  100 CONTINUE
      NE=0
      DO 130 K=1,NG
      DO 130 J=1,NCOMP
      NE=NE+1
      DO 130 L=1,NF
      CZ(L)=CZ(L)+OPCOST(J,K)*BIF(NE,L)
  130 CONTINUE
      NE=0
      DO 150 K=1,NPRO
      DO 150 J=1,NCOMP
      NE=NE+1
      DO 150 L=1,NF
      CZ(L)=CZ(L)-VALPRO(J,K)*PBIF(NE,L)
  150 CONTINUE
      WRITE(6,2)

C     THE JJTH PROCESS AND IMPLIED CONSTRAINT HAS THE FORM
C     RHS(JJ)  (.LE. , .EQ. , OR .GE.)
C        SUM (K=1,NF) CREDVA(J,K)*FX(K)
C         + SUM ((L=1,NCOMP),K=1,NG) CINS(J,L,K)*G(L,K)
C         + SUM ((L=1,NCOMP),K=1,NPRO) CPROD(J,L,K)*P(L,K)
C                (.GE. , .EQ. , OR .LE.)  RNGE(JJ)
C     OR
C     RHS(JJ) (.LE.,.EQ., OR.GE.) SUM (K=1,NF) CC(JJ,K)*X(K)
C           (.GE., .EQ., OR .LE.) RANGE(JJ)
C     WHERE
C       JJ=1,NCON
C
C     CALCULATION OF CC(JJ,N)
C
```

```
      C     SUBROUTINE SFCBDP  ...(CONT'D)
      IF(NCON) 251,251,195
  195 DO 230 J=1,NCON
      M=M+1
      DO 200 K=1,NF
      CC(M,K)=CREDVA(J,K)
  200 CONTINUE
      NV=0
      DO 210 K=1,NG
      DO 210 L=1,NCOMP
      NV=NV+1
      IF(ABS(CINS(J,L,K)).LT.0.00001) GOTO 210
      DO 209 LL=1,NF
      CC(M,LL)=CC(M,LL)+BIF(NV,LL)*CINS(J,L,K)
  209 CONTINUE
  210 CONTINUE
      NV=0
      DO 220 K=1,NPRO
      DO 220 L=1,NCOMP
      NV=NV+1
      IF(ABS(CPROD(J,L,K)).LT.0.00001) GOTO 220
      DO 219 LL=1,NF
      CC(M,LL)=CC(M,LL)+PBIF(NV,LL)*CPROD(J,L,K)
  219 CONTINUE
  220 CONTINUE
  230 CONTINUE
C     RHS'S CONSTRAINT TYPES AND RANGES
C
  240 DO 250 J=1,M
      NSIG(J)=NCODE(J)
      RANGE(J)=RNGE(J)
      R(J)=RHS(J)
  250 CONTINUE
      WRITE(6,3)
C     THE BOUNDS ON THE FX(J) ARE ENTERED IN THE BOUNDS
C     VECTOR.
C
  251 IF(NB) 300,300,255
  255 CONTINUE
      DO 260 J=1,NF
      BOUND(J)=FBND(J,1)
      NBT(J)=FBND(J,2)
  260 CONTINUE
C     BOUNDS ON G(J,K),P(J,K) ARE EXPRESSED AS ADDITIONAL
C     CONSTRAINTS IN TERMS OF FX(K).
C
```

```
      C     SUBROUTINE SFCBDP  ...(CONT'D)
      NV=0
      DO 270 J=1,NG
      DO 270 K=1,NCOMP
      NV=NV+1
      IT=GBND(K,J,2)+1
      GOTO(270,261,262,263,270,270,270),IT
  261 NSIG(M+1)=-1
      GOTO 265
  262 NSIG(M+1)=1
      GOTO 265
  263 NSIG(M+1)=0
  265 M=M+1
      R(M)=GBND(K,J,1)
      RANGE(M)=0.
      DO 266 L=1,NF
      CC(M,L)=BIF(NV,L)
  266 CONTINUE
  270 CONTINUE
      NV=0
      DO 280 J=1,NPRO
      DO 280 K=1,NCOMP
      NV=NV+1
      IT=PBND(K,J,2)+1
      GOTO(280,271,272,273,280,280,280),IT
  271 NSIG(M+1)=-1
      GOTO 275
  272 NSIG(M+1)=1
      GOTO 275
  273 NSIG(M+1)=0
  275 M=M+1
      R(M)=PBND(K,J,1)
      RANGE(M)=0.
      DO 276 L=1,NF
      CC(M,L)=PBIF(NV,L)
  276 CONTINUE
  280 CONTINUE
      WRITE(6,4)
  300 WRITE(6,5)
      RETURN
      END
```

MPS PROGRAM E.P.

EXPANDED PROBLEM

```
        PROGRAM
        TITLE('BUTADIENE AREA L.P. - EXPANDED PROBLEM')
*
*       THIS PROGRAM SOLVES THE EXPANDED FORM OF THE L.P.
*       PROBLEM USING DATA GENERATED BY A FORTRAN PROGRAM.
*
        INITIALZ
        XCLOCKSW=0
        XFREQINV=20
        MOVE(XDATA,'SFEBDP')
        MOVE(XPBNAME,'SFEBDP')
        XPRICE=1
        CONVERT
        SETUP('RANGE','RA.BDP.1','BOUND','BND.BDP1','MIN')
        MOVE(XOBJ,'COSTS')
        MOVE(XRHS,'RHBDP')
        PICTURE
        DUAL
        PRIMAL
        SOLUTION
        RANGE
        EXIT
        PEND
```

MPS PROGRAM R.P.

REDUCED PROBLEM

```
          PROGRAM
          TITLE('BUTADIENE AREA L.P. - REDUCED PROBLEM')
*
*         THIS PROGRAM SOLVES THE REDUCED FORM OF THE L.P.
*         PROBLEM USING DATA GENERATED BY A FORTRAN PROGRAM.
*
          INITIALZ
          XCLOCKSW=0
          XFREQINV=20
          MOVE(XDATA,'SFCBDP')
          MOVE(XPBNAME,'SFCBDP')
          XPRICE=1
          CONVERT
          SETUP('RANGE','RA.BDP.1','BOUND','BND.BDP.','MIN')
          MOVE(XOBJ,'COSTS')
          MOVE(XRHS,'RHBDP')
          PRIMAL
          SOLUTION
          RANGE
          EXIT
          PEND
```

2. TABLES

Input data and O.S. control cards for MPS/360 operation appear in the following tables. For a list of tables, see the title page.

TABLE B - 1

MPSDAT INPUT DATA

EXPANDED PROBLEM

SFEBDP
CREDVA1
CINTS1
CPROD1
MAT.FX8
MAT.FX9
MAT.FX10
MAT.FX11
RG1.1
RG2.1
RG3.1
RG4.1
RG1.2
RG2.2
RG3.2
RG4.2
RG1.3
RG2.3
RG3.3
RG4.3
RG1.4
RG2.4
RG3.4
RG4.4
RG1.5
RG2.5
RG3.5
RG4.5
RG1.6
RG2.6
RG3.6
RG4.6
RG1.7
RG2.7
RG3.7
RG4.7
RG1.8
RG2.8
RG3.8
RG4.8
RG1.9
RG2.9
RG3.9
RG4.9
RG1.10
RG2.10
RG3.10

TABLE B - 1 ...CONT'D

RG4.10
RG1.11
RG2.11
RG3.11
RG4.11
RG1.12
RG2.12
RG3.12
RG4.12
RG1.13
RG2.13
RG3.13
RG4.13
RG1.14
RG2.14
RG3.14
RG4.14
RG1.15
RG2.15
RG3.15
RG4.15
RP1.1
RP2.1
RP3.1
RP4.1
RP1.2
RP2.2
RP3.2
RP4.2
RP1.3
RP2.3
RP3.3
RP4.3
RP1.4
RP2.4
RP3.4
RP4.4
RP1.5
RP2.5
RP3.5
RP4.5
RP1.6
RP2.6
RP3.6
RP4.6
FX1
FX2
FX3
FX4
FX5
FX6
FX7

TABLE B - 1 ...CONT'D

FX8
FX9
FX10
FX11
GG1.1
GG2.1
GG3.1
GG4.1
GG1.2
GG2.2
GG3.2
GG4.2
GG1.3
GG2.3
GG3.3
GG4.3
GG1.4
GG2.4
GG3.4
GG4.4
GG1.5
GG2.5
GG3.5
GG4.5
GG1.6
GG2.6
GG3.6
GG4.6
GG1.7
GG2.7
GG3.7
GG4.7
GG1.8
GG2.8
GG3.8
GG4.8
GG1.9
GG2.9
GG3.9
GG4.9
GG1.10
GG2.10
GG3.10
GG4.10
GG1.11
GG2.11
GG3.11
GG4.11
GG1.12
GG2.12
GG3.12
GG4.12

TABLE B - 1 ...CONT'D

GG1.13
GG2.13
GG3.13
GG4.13
GG1.14
GG2.14
GG3.14
GG4.14
GG1.15
GG2.15
GG3.15
GG4.15
P1.1
P2.1
P3.1
P4.1
P1.2
P2.2
P3.2
P4.2
P1.3
P2.3
P3.3
P4.3
P1.4
P2.4
P3.4
P4.4
P1.5
P2.5
P3.5
P4.5
P1.6
P2.6
P3.6
P4.6
COSTS
RHBDP
RA.BDP.1
BND.BDP1

TABLE B - 2

MPSDAT INPUT DATA

REDUCED PROBLEM

SFCBDP
CREDVA1
CINTS1
CPROD1
MAT.FX8
MAT.FX9
MAT.FX10
MAT.FX11
RG2.1
RG1.2
RG1.3
RG2.10
RG4.5
RG4.6
FX1
FX2
FX3
FX4
FX5
FX6
FX7
FX8
FX9
FX10
FX11
COSTS
RHBDP
RA.BDP.1
BND.BDP.1

TABLE B - 3

DATA GENERATED FOR MPS/360

EXPANDED PROBLEM

```
NAME          SFEBDP
ROWS
 N  COSTS
 L  CREDVA1
 L  CINTS1
 G  CPROD1
 L  MAT.FX8
 L  MAT.FX9
 L  MAT.FX10
 L  MAT.FX11
 E  RG1.1
 E  RG2.1
 E  RG3.1
 E  RG4.1
 E  RG1.2
 E  RG2.2
 E  RG3.2
 E  RG4.2
 E  RG1.3
 E  RG2.3
 E  RG3.3
 E  RG4.3
 E  RG1.4
 E  RG2.4
 E  RG3.4
 E  RG4.4
 E  RG1.5
 E  RG2.5
 E  RG3.5
 E  RG4.5
 E  RG1.6
 E  RG2.6
 E  RG3.6
 E  RG4.6
 E  RG1.7
 E  RG2.7
 E  RG3.7
 E  RG4.7
 E  RG1.8
 E  RG2.8
 E  RG3.8
 E  RG4.8
 E  RG1.9
 E  RG2.9
 E  RG3.9
 E  RG4.9
 E  RG1.10
```

TABLE B - 3 ...CONT'D

```
 E   RG2.10
 E   RG3.10
 E   RG4.10
 E   RG1.11
 E   RG2.11
 E   RG3.11
 E   RG4.11
 E   RG1.12
 E   RG2.12
 E   RG3.12
 E   RG4.12
 E   RG1.13
 E   RG2.13
 E   RG3.13
 E   RG4.13
 E   RG1.14
 E   RG2.14
 E   RG3.14
 E   RG4.14
 E   RG1.15
 E   RG2.15
 E   RG3.15
 E   RG4.15
 E   RP1.1
 E   RP2.1
 E   RP3.1
 E   RP4.1
 E   RP1.2
 E   RP2.2
 E   RP3.2
 E   RP4.2
 E   RP1.3
 E   RP2.3
 E   RP3.3
 E   RP4.3
 E   RP1.4
 E   RP2.4
 E   RP3.4
 E   RP4.4
 E   RP1.5
 E   RP2.5
 E   RP3.5
 E   RP4.5
 E   RP1.6
 E   RP2.6
 E   RP3.6
 E   RP4.6
COLUMNS
    FX1       COSTS         0.020000
    FX1       RG1.1        -0.400000
    FX1       RG2.1        -0.250000
    FX1       RG3.1        -0.350000
```

TABLE B - 3 ...CONT'D

FX2	COSTS	0.030000
FX2	RG1.3	-0.920000
FX2	RG2.3	-0.030000
FX2	RG3.3	-0.040000
FX2	RG4.3	-0.010000
FX3	COSTS	0.048000
FX3	RG1.4	-0.600000
FX3	RG3.4	-0.100000
FX3	RG4.4	-0.300000
FX4	COSTS	0.100000
FX4	CREDVA1	1.00000
FX4	RG1.4	-0.080000
FX4	RG2.4	-0.010000
FX4	RG3.4	-0.010000
FX4	RG4.4	-0.900000
FX5	COSTS	0.100000
FX5	CREDVA1	1.00000
FX5	RG1.5	-0.080000
FX5	RG2.5	-0.010000
FX5	RG3.5	-0.010000
FX5	RG4.5	-0.900000
FX6	COSTS	0.038000
FX6	RG1.6	-0.300000
FX6	RG2.6	-0.300000
FX6	RG4.6	-0.400000
FX7	COSTS	0.110000
FX7	RG1.7	-0.100000
FX7	RG4.7	-0.900000
FX8	COSTS	-0.034000
FX8	MAT.FX8	1.00000
FX8	RG2.9	1.00000
FX9	COSTS	-0.084000
FX9	MAT.FX9	1.00000
FX9	RG2.1	1.00000
FX10	COSTS	-0.033000
FX10	MAT.FX10	1.00000
FX10	RG2.10	1.00000
FX11	COSTS	-0.034000
FX11	MAT.FX11	1.00000
FX11	RG2.1	1.00000
GG1.1	RG1.1	1.00000
GG1.1	RG1.2	-1.00000
GG2.1	COSTS	0.032000
GG2.1	RG2.1	1.00000
GG2.1	RG2.2	-0.150000
GG2.1	RG2.8	-0.850000
GG3.1	RG3.1	1.00000
GG3.1	RG3.2	-1.00000
GG4.1	RG4.1	1.00000
GG4.1	RG4.2	-1.00000
GG1.2	COSTS	0.032000
GG1.2	CINTS1	1.00000

TABLE B - 3 ...CONT'D

GG1.2	RG1.2	1.00000
GG1.2	RG1.3	-0.920000
GG1.2	RP1.1	-0.080000
GG2.2	CINTS1	1.00000
GG2.2	RG2.2	1.00000
GG2.2	RG2.3	-0.900000
GG2.2	RP2.1	-0.100000
GG3.2	CINTS1	1.00000
GG3.2	RG3.2	1.00000
GG3.2	RG3.3	-0.150000
GG3.2	RP3.1	-0.850000
GG4.2	CINTS1	1.00000
GG4.2	RG4.2	1.00000
GG4.2	RG4.3	-0.980000
GG4.2	RP4.1	-0.020000
GG1.3	RG1.3	1.00000
GG1.3	RG1.4	-0.600000
GG1.3	RG4.4	-0.360000
GG2.3	RG2.3	1.00000
GG2.3	RG2.4	-0.970000
GG3.3	RG3.3	1.00000
GG3.3	RG3.4	-0.900000
GG4.3	RG4.3	1.00000
GG4.3	RG4.4	-0.900000
GG1.4	RG1.4	1.00000
GG1.4	RG1.11	-0.950000
GG1.4	RP1.2	-0.050000
GG2.4	RG2.4	1.00000
GG2.4	RG2.11	-0.950000
GG2.4	RP2.2	-0.050000
GG3.4	RG3.4	1.00000
GG3.4	RG3.11	-0.950000
GG3.4	RP3.2	-0.050000
GG4.4	COSTS	0.028000
GG4.4	RG4.4	1.00000
GG4.4	RG4.11	-0.950000
GG4.4	RP4.2	-0.050000
GG1.5	RG1.5	1.00000
GG1.5	RG1.12	-1.00000
GG2.5	RG2.5	1.00000
GG2.5	RG2.12	-1.00000
GG3.5	RG3.5	1.00000
GG3.5	RG3.12	-1.00000
GG4.5	COSTS	0.007600
GG4.5	RG4.5	1.00000
GG4.5	RG4.12	-0.050000
GG4.5	RP4.3	-0.950000
GG1.6	RG1.6	1.00000
GG1.6	RG1.7	-0.200000
GG1.6	RG1.14	-0.800000
GG2.6	RG2.6	1.00000
GG2.6	RG2.7	-0.200000

TABLE B - 3 ...CONT'D

GG2.6	RG2.14	-0.800000
GG3.6	RG3.6	1.00000
GG3.6	RG3.14	-1.00000
GG4.6	COSTS	0.006400
GG4.6	RG4.6	1.00000
GG4.6	RG4.7	-0.950000
GG4.6	RG4.14	-0.050000
GG1.7	RG1.7	1.00000
GG1.7	RG1.15	-1.00000
GG2.7	RG2.7	1.00000
GG2.7	RG2.15	-1.00000
GG3.7	RG3.7	1.00000
GG3.7	RP3.4	-1.00000
GG4.7	COSTS	0.002900
GG4.7	RG4.7	1.00000
GG4.7	RG4.15	-0.060000
GG4.7	RP4.4	-0.950000
GG1.8	RG1.8	1.00000
GG2.8	MAT.FX8	-0.800000
GG2.8	RG2.8	1.00000
GG2.8	RG2.9	-0.800000
GG2.8	RP2.5	-0.100000
GG3.8	RG3.8	1.00000
GG4.8	RG4.8	1.00000
GG1.9	RG1.9	1.00000
GG2.9	MAT.FX9	-0.100000
GG2.9	MAT.FX10	-0.900000
GG2.9	RG2.1	-0.100000
GG2.9	RG2.9	1.00000
GG2.9	RG2.10	-0.900000
GG3.9	RG3.9	1.00000
GG4.9	RG4.9	1.00000
GG1.10	RG1.10	1.00000
GG2.10	COSTS	0.052000
GG2.10	MAT.FX11	-0.050000
GG2.10	RG2.1	-0.100000
GG2.10	RG2.10	1.00000
GG2.10	RP2.6	-0.850000
GG3.10	RG3.10	1.00000
GG4.10	RG4.10	1.00000
GG1.11	RG1.5	-0.500000
GG1.11	RG1.6	-0.500000
GG1.11	RG1.11	1.00000
GG2.11	RG2.5	-0.500000
GG2.11	RG2.6	-0.500000
GG2.11	RG2.11	1.00000
GG3.11	RG3.5	-0.500000
GG3.11	RG3.6	-0.500000
GG3.11	RG3.11	1.00000
GG4.11	RG4.5	-0.500000
GG4.11	RG4.6	-0.500000
GG4.11	RG4.11	1.00000

TABLE B - 3 ...CONT'D

GG1.12	RG1.3	-0.800000
GG1.12	RG1.12	1.00000
GG1.12	RG1.13	-0.200000
GG2.12	RG2.3	-0.800000
GG2.12	RG2.12	1.00000
GG2.12	RG2.13	-0.200000
GG3.12	RG3.3	-0.800000
GG3.12	RG3.12	1.00000
GG3.12	RG3.13	-0.200000
GG4.12	RG4.3	-0.800000
GG4.12	RG4.12	1.00000
GG4.12	RG4.13	-0.200000
GG1.13	RG1.1	-0.900000
GG1.13	RG1.2	-0.100000
GG1.13	RG1.13	1.00000
GG2.13	RG2.1	-0.900000
GG2.13	RG2.2	-0.100000
GG2.13	RG2.13	1.00000
GG3.13	RG3.1	-0.900000
GG3.13	RG3.2	-0.100000
GG3.13	RG3.13	1.00000
GG4.13	RG4.1	-0.900000
GG4.13	RG4.2	-0.100000
GG4.13	RG4.13	1.00000
GG1.14	RG1.1	-0.650000
GG1.14	RG1.3	-0.350000
GG1.14	RG1.14	1.00000
GG2.14	RG2.1	-0.650000
GG2.14	RG2.3	-0.350000
GG2.14	RG2.14	1.00000
GG3.14	RG3.1	-0.650000
GG3.14	RG3.3	-0.350000
GG3.14	RG3.14	1.00000
GG4.14	RG4.1	-0.650000
GG4.14	RG4.3	-0.350000
GG4.14	RG4.14	1.00000
GG1.15	RG1.1	-0.050000
GG1.15	RG1.3	-0.950000
GG1.15	RG1.15	1.00000
GG2.15	RG2.1	-0.050000
GG2.15	RG2.3	-0.950000
GG2.15	RG2.15	1.00000
GG3.15	RG3.1	-0.050000
GG3.15	RG3.3	-0.950000
GG3.15	RG3.15	1.00000
GG4.15	RG4.1	-0.050000
GG4.15	RG4.3	-0.950000
GG4.15	RG4.15	1.00000
P1.1	RP1.1	1.00000
P2.1	RP2.1	1.00000
P3.1	COSTS	-0.029000
P3.1	RP3.1	1.00000

TABLE B - 3 ...CONT'D

```
    P4.1      RP4.1          1.00000
    P1.2      COSTS         -0.012000
    P1.2      RP1.2          1.00000
    P2.2      COSTS         -0.012000
    P2.2      RP2.2          1.00000
    P3.2      COSTS         -0.012000
    P3.2      RP3.2          1.00000
    P4.2      COSTS         -0.012000
    P4.2      RP4.2          1.00000
    P1.3      RP1.3          1.00000
    P2.3      RP2.3          1.00000
    P3.3      RP3.3          1.00000
    P4.3      COSTS         -0.120000
    P4.3      CPROD1         1.00000
    P4.3      RP4.3          1.00000
    P1.4      RP1.4          1.00000
    P2.4      RP2.4          1.00000
    P3.4      RP3.4          1.00000
    P4.4      COSTS         -0.120000
    P4.4      CPROD1         1.00000
    P4.4      RP4.4          1.00000
    P1.5      RP1.5          1.00000
    P2.5      COSTS         -0.026000
    P2.5      RP2.5          1.00000
    P3.5      RP3.5          1.00000
    P4.5      RP4.5          1.00000
    P1.6      RP1.6          1.00000
    P2.6      COSTS         -0.300000
    P2.6      RP2.6          1.00000
    P3.6      RP3.6          1.00000
    P4.6      RP4.6          1.00000
RHS
    RHBDP     CREDVA1        0.014000
    RHBDP     CINTS1         1.05000
    RHBDP     CPROD1         0.060000
RANGES
    RA.BDP.1  CREDVA1        0.0
    RA.BDP.1  CPROD1         0.280000
BOUNDS
 UP BND.BDP1  FX1            0.960000
 UP BND.BDP1  FX2            0.105000
 UP BND.BDP1  FX3            0.027000
 UP BND.BDP1  FX6            0.038000
 UP BND.BDP1  FX7            0.045000
 UP BND.BDP1  GG2.1          0.230000
 UP BND.BDP1  GG1.2          0.660000
 UP BND.BDP1  GG1.3          1.15000
 UP BND.BDP1  GG4.5          0.260000
 UP BND.BDP1  GG4.6          0.260000
 UP BND.BDP1  GG2.10         0.140000
ENDATA
```

TABLE B - 4

DATA GENERATED FOR MPS/360

REDUCED PROBLEM

```
NAME            SFCBDP
ROWS
 N  COSTS
 L  CREDVA1
 L  CINTS1
 G  CPROD1
 L  MAT.FX8
 L  MAT.FX9
 L  MAT.FX10
 L  MAT.FX11
 L  RG2.1
 L  RG1.2
 L  RG1.3
 L  RG2.10
 L  RG4.5
 L  RG4.6
COLUMNS
    FX1         COSTS           -0.025102
    FX1         CINTS1            1.02050
    FX1         CPROD1           0.279661
    FX1         MAT.FX8         -0.223046
    FX1         MAT.FX9         -0.022305
    FX1         MAT.FX10        -0.200742
    FX1         MAT.FX11        -0.010037
    FX1         RG2.1            0.328009
    FX1         RG1.2            0.571416
    FX1         RG1.3            0.823921
    FX1         RG2.10           0.150964
    FX1         RG4.5            0.150964
    FX1         RG4.6            0.200742
    FX2         COSTS           -0.017387
    FX2         CINTS1           0.483417
    FX2         CPROD1           0.707639
    FX2         MAT.FX8         -0.021537
    FX2         MAT.FX9         -0.002154
    FX2         MAT.FX10        -0.019383
    FX2         MAT.FX11        -0.000969
    FX2         RG2.1            0.031672
    FX2         RG1.2            0.428542
    FX2         RG1.3             2.05980
    FX2         RG2.10           0.381991
    FX2         RG4.5            0.381991
    FX2         RG4.6            0.019383
    FX3         COSTS            0.003633
    FX3         CINTS1           0.571983
    FX3         CPROD1           0.703378
    FX3         RG1.2            0.465807
```

TABLE B - 4 ...CONT'D

FX3	RG1.3	1.23892
FX3	RG2.10	0.379691
FX3	RG4.5	0.379691
FX4	COSTS	0.028397
FX4	CREDVA1	1.00000
FX4	CINTS1	0.095049
FX4	CPROD1	0.904638
FX4	MAT.FX8	-0.007401
FX4	MAT.FX9	-0.000740
FX4	MAT.FX10	-0.006661
FX4	MAT.FX11	-0.000333
FX4	RG2.1	0.010884
FX4	RG1.2	0.062108
FX4	RG1.3	0.165189
FX4	RG2.10	0.488333
FX4	RG4.5	0.488333
FX4	RG4.6	0.006661
FX5	COSTS	-0.003224
FX5	CREDVA1	1.00000
FX5	CINTS1	0.074778
FX5	CPROD1	0.952855
FX5	MAT.FX8	-0.007478
FX5	MAT.FX9	-0.000748
FX5	MAT.FX10	-0.006730
FX5	MAT.FX11	-0.000337
FX5	RG2.1	0.010998
FX5	RG1.2	0.052668
FX5	RG1.3	0.176247
FX5	RG2.10	0.952823
FX5	RG4.5	0.052824
FX5	RG4.6	0.006730
FX6	COSTS	-0.046857
FX6	CINTS1	0.371095
FX6	CPROD1	0.615397
FX6	MAT.FX8	-0.243080
FX6	MAT.FX9	-0.024308
FX6	MAT.FX10	-0.218772
FX6	MAT.FX11	-0.010939
FX6	RG2.1	0.357471
FX6	RG1.2	0.292817
FX6	RG1.3	0.643195
FX6	RG2.10	0.137326
FX6	RG4.5	0.537326
FX6	RG4.6	0.218772
FX7	COSTS	0.001879
FX7	CINTS1	0.057068
FX7	CPROD1	0.976467
FX7	RG1.2	0.051394
FX7	RG1.3	0.222996
FX7	RG2.10	0.065570
FX7	RG4.5	0.065570
FX8	COSTS	0.172292

TABLE B - 4 ...CONT'D

```
    FX8       CINTS1       -0.038156
    FX8       MAT.FX8       1.16951
    FX8       MAT.FX9       0.116951
    FX8       MAT.FX10      1.05256
    FX8       MAT.FX11      0.052628
    FX8       RG2.1        -0.249287
    FX8       RG4.6        -1.05256
    FX9       COSTS         0.040170
    FX9       CINTS1       -0.200820
    FX9       MAT.FX8       0.892184
    FX9       MAT.FX9       1.08922
    FX9       MAT.FX10      0.802967
    FX9       MAT.FX11      0.040148
    FX9       RG2.1        -1.31204
    FX9       RG4.6        -0.802966
    FX10      COSTS         0.182417
    FX10      CINTS1       -0.020082
    FX10      MAT.FX8       0.089218
    FX10      MAT.FX9       0.008922
    FX10      MAT.FX10      1.08030
    FX10      MAT.FX11      0.054015
    FX10      RG2.1        -0.131204
    FX10      RG4.6        -1.08030
    FX11      COSTS         0.090170
    FX11      CINTS1       -0.200820
    FX11      MAT.FX8       0.892184
    FX11      MAT.FX9       0.089218
    FX11      MAT.FX10      0.802967
    FX11      MAT.FX11      1.04015
    FX11      RG2.1        -1.31204
    FX11      RG4.6        -0.802966
RHS
    RHBDP     CREDVA1       0.014000
    RHBDP     CINTS1        1.05000
    RHBDP     CPROD1        0.060000
    RHBDP     RG2.1         0.230000
    RHBDP     RG1.2         0.660000
    RHBDP     RG1.3         1.15000
    RHBDP     RG2.10        0.260000
    RHBDP     RG4.5         0.260000
    RHBDP     RG4.6         0.140000
RANGES
    RA.BDP.1  CREDVA1       0.0
    RA.BDP.1  CPROD1        0.280000
BOUNDS
 UP BND.BDP.  FX1           0.960000
 UP BND.BDP.  FX2           0.105000
 UP BND.BDP.  FX3           0.027000
 UP BND.BDP.  FX6           0.038000
 UP BND.BDP.  FX7           0.045000
ENDATA
```

TABLE B - 5

O.S. CONTROL CARDS REQUIRED

TYPICAL MPS/360 RUN

```
//LCBIRP JOB (CPG1,2,2),'REDUCED PROB.'
//JOBLIB   DD   DSNAME=SYS1.LPLIB,DISP=SHR
//CPC EXEC PGM=COMPILER
//SCRATCH1   DD   UNIT=SYSDA,SPACE=(TRK,(2,2))
//SCRATCH2   DD   UNIT=SYSDA,SPACE=(TRK,(2,2))
//SCRATCH3   DD   UNIT=SYSDA,SPACE=(TRK,(2,2))
//SCRATCH4   DD   UNIT=SYSDA,SPACE=(TRK,(2,2))
//SYSMLCP   DD   UNIT=SYSDA,SPACE=(TRK,(2,2)),DISP=(NEW,PASS)
//SYSPRINT   DD   SYSOUT=A
//SYSIN   DD   *

                        .
                        .
                   MPS PROGRAM
                        .
                        .

/*
//STEP2 EXEC PGM=EXECUTOR,COND=(0,NE,CPC)
//SYSMLCP   DD   DSNAME=*.CPC.SYSMLCP,DISP=(OLD,DELETE)
//MATRIX1   DD   UNIT=SYSDA,SPACE=(CYL,(5),,CONTIG)
//ETA1      DD   UNIT=SYSDA,SPACE=(CYL,(5),,CONTIG)
//SCRATCH1 DD   UNIT=SYSDA,SPACE=(CYL,(5),,CONTIG)
//SCRATCH2 DD   UNIT=SYSDA,SPACE=(CYL,(5),,CONTIG)
//PROBFILE DD   UNIT=SYSDA,SPACE=(CYL,(5),,CONTIG)
//NEWPFILE DD   UNIT=SYSDA,SPACE=(CYL,(5),,CONTIG)
//SYSPRINT   DD   SYSOUT=A
//SYSIN   DD   *

                        .
                        .
                    MPS DATA
                        .
                        .

/*
/*
```

3. DOCUMENTATION

3.1 Procedure

Linear programming was used to find optimal solutions to the specific optimization model for the butadiene process. The form of the model transformation equations used is slightly different (in equation order only) but will be documented by comment cards in the appropriate printouts.

The procedure used consists of two basic steps: generation of the optimization model in a form suitable for input to MPS/360, and solution of the optimization problem by MPS/360. The first involves standard Fortran programming and job control; the second requires a knowledge of MPS/360 procedures and appropriate job control language. A typical set of O.S. control cards required for an MPS/360 run are listed in table B-5.

Details of the procedure, including definition of variables, are documented by comment cards in the printout section: Subroutine INPUT (NB,NR) is documented in Appendix C.

3.2 Input-Output

The Fortran model generation step requires input of the model data and MPSDAT data. The model data required appears in Appendix C, table C-1, MPSDAT data are listed

in tables B-1 and B-2. The Fortran programs generate MPS/360 data defining the appropriate optimization problem. These data are listed in tables B-3 and B-4.

The generated MPS/360 data are read (for the programs documented here, from cards), and the L.P. problem solved. The MPS/360 printout is extensive and not included here. The results are summarized in table 9.

3.3 Comments

MPS/360 proved to be an easy-to-use, efficient and accurate procedure. Additional features of the package were required to continue analysis but were not implemented at the time they were needed. For this reason, the remainder of the analysis was done using a simple two-phase simplex algorithm written in Fortran. However, there is no comparison in accuracy or efficiency; every effort should be made to use MPS/360 or a comparable procedure for large problems.

APPENDIX C

MODEL DATA INPUT PROCEDURE

Table of Contents

LIST OF TABLES

SUBROUTINE INPUT

```
      SUBROUTINE INPUT
C     SUBROUTINE INPUT READS IN DATA NECESSARY TO GENERATE
C     THE COEFFICIENTS OF THE LINEAR PROGRAMMING PROBLEM.
C
      COMMON A(4,15,15),Y(4,15,11),D(4,6,15),CPROD(10,4,6),
     1CINS(10,4,15),CREDVA(10,11),FDCOST(4,11),
     1OPCOST(4,15),VALPRO(4,6),CON(4),RHS(10),RNGE(10),
     1GBND(4,15,2),PBND(4,6,2),FBND(11,2),SEL,NCODE(10),
     1NCOMP,NPRO,NG,NF,NCON,NB,NR
      REAL DNAME(16,2)/'CON','RHS','NCOD','RNGE','FBND',
     1'OPCO','FDCO','VALP','CRED','A','CINS','CPRO',
     2'Y','D','GBND','PBND',' ',' ','E',' ',' ','ST','ST',
     3'RO','VA',' ',' ','D',' ',' ',' ',' '/
    1 FORMAT(I10)
    2 FORMAT(15X,2A4,12X,I3)
    3 FORMAT(G14.6)
    4 FORMAT(1H1,//////,' ',12X,'INPUT OF DATA FOR THE ',
     1'STANDARD L.P. PROBLEM')
    6 FORMAT(1H0,12X,'THE ARRAYS READ IN ARE -'/1H0,12X,
     1'ARRAY NAME',10X,'CODE NO.')
    7 FORMAT(1H0,12X,'INPUT FINISHED')
    8 FORMAT(1H0,12X,'CATALYST SELECTIVITY =',F4.2)
    9 FORMAT(1H0,12X,'NO. OF COMPONENTS =',I2)
   11 FORMAT(1H0,12X,'NO. OF UNITS =',I3)
   12 FORMAT(1H0,12X,'NO. OF FRESH FEED STREAMS =',I3)
   13 FORMAT(1H0,12X,'NO. OF INEQUALITY CONSTRAINTS =',I3)
   14 FORMAT(1H0,12X,'NB=' ,I2,'      NR=',I2/)
   15 FORMAT(1H0,12X,'NO. OF PRODUCT STREAMS =',I2)
      WRITE(6,4)
      I2=2
      I4=4
      I6=6
      I10=10
      I11=11
      I15=15
      I100=100
C     THE VARIABLES ARE READ IN.
C
      READ(5,3) SEL
      READ(5,1) NCOMP
      READ(5,1) NPRO
      READ(5,1) NG
      READ(5,1) NF
      READ(5,1) NCON
      READ(5,1) NB
      READ(5,1) NR
```

SUBROUTINE INPUT ...(CONT'D)

```
      WRITE(6,8) SEL
      WRITE(6,9) NCOMP
      WRITE(6,15) NPRO
      WRITE(6,11) NG
      WRITE(6,12) NF
      WRITE(6,13) NCON
      WRITE(6,14) NB,NR
      WRITE(6,6)

C     THE ARRAYS ARE READ IN.
C     BECAUSE THE ARRAYS TO BE READ IN ARE SPARSE, ONLY THE
C     NON-ZERO ELEMENTS ARE READ IN.  THIS INVOLVES THE USE
C     OF SEVERAL INPUT ROUTINES.
C           IMINP1 - FOR INTEGER ARRAYS OF DIMENSION 1
C           MINP1  - FOR REAL ARRAYS OF DIMENSION 1
C           MINP2  - FOR REAL ARRAYS OF DIMENSION 2
C           MINP3  - FOR REAL ARRAYS OF DIMENSION 3
C     EACH ARRAY IS ASSIGNED A CODE NO. - NT.  THE CODE NO.
C     IS READ BY INPUT.  CONTROL IS THEN TRANSFERRED TO
C     THE APPROPRIATE SUBROUTINE.  A CODE NO. OF 0 INDICATES
C     THAT ALL ARRAYS HAVE BEEN READ.  THE MINP SUBROUTINE
C     USED INITIALIZES THE ARRAY TO ZERO AND THEN READS IN
C     THE ARRAY ELEMENTS AND THEIR INDICES.  AN INDEX OF 0
C     INDICATES THAT ALL DATA FOR THAT ARRAY HAVE BEEN READ.
C

 1000 READ(5,1) NT
      IF(NT) 2000,2000,1500
 1500 WRITE(6,2)(DNAME(NT,J),J=1,2),NT
      GOTO(10,20,30,40,50,60,70,80,90,100,110,120,130,140,
     1150,160),NT
   10 CALL MINP1(CON,I4,&1000)
   20 CALL MINP1(RHS,I10,&1000)
   30 CALL IMINP1(NCODE,I10,&1000)
   40 CALL MINP1(RNGE,I10,&1000)
   50 CALL MINP2(FBND,I11,I2,&1000)
   60 CALL MINP2(OPCOST,I4,I15,&1000)
   70 CALL MINP2(FDCOST,I4,I11,&1000)
   80 CALL MINP2(VALPRO,I4,I6,&1000)
   90 CALL MINP2 (CREDVA,I10,I11,&1000)
  100 CALL MINP3( A,I4,I15,I15,&1000)
  110 CALL MINP3(CINS,I10,I4,I15,&1000)
  120 CALL MINP3(CPROD,I10,I4,I6,&1000)
  130 CALL MINP3(Y,I4,I15,I11,&1000)
  140 CALL MINP3(D,I4,I6,I15,&1000)
  150 CALL MINP3(GBND,I4,I15,I2,&1000)
  160 CALL MINP3(PBND,I4,I6,I2,&1000)
 2000 CONTINUE
      WRITE(6,7)
      RETURN
```

SUBROUTINE INPUT ...(CONT'D)

```
END
```

SUBROUTINE INPUT

```
      SUBROUTINE INPUT(NB,NR)
C     SUBROUTINE INPUT READS IN DATA NECESSARY TO GENERATE
C     THE COEFFICIENTS OF THE LINEAR PROGRAMMING PROBLEM.
C
C     THIS VERSION OF INPUT IS USED ONLY IN THE MPS DATA
C     GENERATION PROGRAMS
C
      COMMON A(4,15,15),Y(4,15,11),D(4,6,15),CPROD(10,4,6),
     1CINS(10,4,15),CREDVA(10,11),FDCOST(4,11),
     1OPCOST(4,15),VALPRO(4,6),CON(4),RHS(10),RNGE(10),
     1GBND(4,15,2),PBND(4,6,2),FBND(11,2),SEL,NCODE(10),
     1NCOMP,NPRO,NG,NF,NCON
      REAL DNAME(16,2)/'CON','RHS','NCOD','RNGE','FBND',
     1'OPCO','FDCO','VALP','CRED','A','CINS','CPRO',
     2'Y','D','GBND','PBND',' ',' ','E',' ',' ','ST','ST',
     3'RO','VA',' ',' ','D',' ',' ',' ',' '/
    1 FORMAT(I10)
    2 FORMAT(15X,2A4,12X,I3)
    3 FORMAT(G14.6)
    4 FORMAT(1H1,//////,' ',12X,'INPUT OF DATA FOR THE ',
     1'STANDARD L.P. PROBLEM')
    6 FORMAT(1H0,12X,'THE ARRAYS READ IN ARE -'/1H0,12X,
     1'ARRAY NAME',10X,'CODE NO.')
    7 FORMAT(1H0,12X,'INPUT FINISHED')
    8 FORMAT(1H0,12X,'CATALYST SELECTIVITY =',F4.2)
    9 FORMAT(1H0,12X,'NO. OF COMPONENTS =',I2)
   11 FORMAT(1H0,12X,'NO. OF UNITS =',I3)
   12 FORMAT(1H0,12X,'NO. OF FRESH FEED STREAMS =',I3)
   13 FORMAT(1H0,12X,'NO. OF INEQUALITY CONSTRAINTS =',I3)
   14 FORMAT(1H0,12X,'NB=' ,I2,'      NR=',I2/)
   15 FORMAT(1H0,12X,'NO. OF PRODUCT STREAMS =',I2)
      WRITE(6,4)
      I2=2
      I4=4
      I6=6
      I11=11
      I10=10
      I15=15
      I100=100
C     THE VARIABLES ARE READ IN.
C
      READ(5,3) SEL
      READ(5,1) NCOMP
      READ(5,1) NPRO
      READ(5,1) NG
      READ(5,1) NF
```

SUBROUTINE INPUT ...(CONT'D)

```
      READ(5,1) NCON
      READ(5,1) NB
      READ(5,1) NR
      WRITE(6,8) SEL
      WRITE(6,9) NCOMP
      WRITE(6,15) NPRO
      WRITE(6,11) NG
      WRITE(6,12) NF
      WRITE(6,13) NCON
      WRITE(6,14) NB,NR
      WRITE(6,6)

C     THE ARRAYS ARE READ IN.
C     BECAUSE THE ARRAYS TO BE READ IN ARE SPARSE, ONLY THE
C     NON-ZERO ELEMENTS ARE READ IN.  THIS INVOLVES THE USE
C     OF SEVERAL INPUT ROUTINES.
C           IMINP1 - FOR INTEGER ARRAYS OF DIMENSION 1
C           MINP1  - FOR REAL ARRAYS OF DIMENSION 1
C           MINP2  - FOR REAL ARRAYS OF DIMENSION 2
C           MINP3  - FOR REAL ARRAYS OF DIMENSION 3
C     EACH ARRAY IS ASSIGNED A CODE NO. - NT.  THE CODE NO.
C     IS READ BY INPUT.  CONTROL IS THEN TRANSFERRED TO
C     THE APPROPRIATE SUBROUTINE.  A CODE NO. OF 0 INDICATES
C     THAT ALL ARRAYS HAVE BEEN READ.  THE MINP SUBROUTINE
C     USED INITIALIZES THE ARRAY TO ZERO AND THEN READS IN
C     THE ARRAY ELEMENTS AND THEIR INDICES.  AN INDEX OF 0
C     INDICATES THAT ALL DATA FOR THAT ARRAY HAVE BEEN READ.
C

 1000 READ(5,1) NT
      IF(NT) 2000,2000,1500
 1500 WRITE(6,2)(DNAME(NT,J),J=1,2),NT
      GOTO(10,20,30,40,50,60,70,80,90,100,110,120,130,140,
     1150,160),NT
   10 CALL MINP1(CON,I4,&1000)
   20 CALL MINP1(RHS,I10,&1000)
   30 CALL IMINP1(NCODE,I10,&1000)
   40 CALL MINP1(RNGE,I10,&1000)
   50 CALL MINP2(FBND,I11,I2,&1000)
   60 CALL MINP2(OPCOST,I4,I15,&1000)
   70 CALL MINP2(FDCOST,I4,I11,&1000)
   80 CALL MINP2(VALPRO,I4,I6,&1000)
   90 CALL MINP2 (CREDVA,I10,I11,&1000)
  100 CALL MINP3( A,I4,I15,I15,&1000)
  110 CALL MINP3(CINS,I10,I4,I15,&1000)
  120 CALL MINP3(CPROD,I10,I4,I6,&1000)
  130 CALL MINP3(Y,I4,I15,I11,&1000)
  140 CALL MINP3(D,I4,I6,I15,&1000)
  150 CALL MINP3(GBND,I4,I15,I2,&1000)
  160 CALL MINP3(PBND,I4,I6,I2,&1000)
```

SUBROUTINE INPUT ...(CONT'D)

```
2000 CONTINUE
     WRITE(6,7)
     RETURN
     END
```

C SUBROUTINE IMINP1

```
C     SUBROUTINE IMINP1
C
C      THIS SUBROUTINE IS USED TO INITIALIZE THE ARRAY A -
C      AN INTEGER ARRAY OF DIMENSION A(IJ).
C
C      FIRST, THE ARRAY IS INITIALIZED TO 0.  THEN THE NON-
C      ZERO ELEMENTS (WITH THEIR INDICES) ARE READ IN AND
C      ENTERED.  AN INDEX OF 0 INDICATES THAT ALL DATA FOR
C      THE ARRAY HAVE BEEN READ.
C
      SUBROUTINE IMINP1(A,IJ,*)
      INTEGER A(IJ),VAL
      DO 10 J=1,IJ
      A(J)=0.0
   10 CONTINUE
   50 READ(5,1) J,VAL
      IF(J) 100,100,60
   60 A(J)=VAL
      GOTO 50
    1 FORMAT(2I10)
  100 RETURN 1
      END
```

C SUBROUTINE MINP1

```
C     SUBROUTINE MINP1

C     THIS SUBROUTINE IS USED TO INITIALIZE THE ARRAY A -
C     A REAL ARRAY OF DIMENSION A(IJ)
C
C     FIRST, THE ARRAY IS INITIALIZED TO 0.  THEN THE NON-
C     ZERO ELEMENTS (WITH THEIR INDICES) ARE READ IN AND
C     ENTERED.  AN INDEX OF 0 INDICATES THAT ALL DATA FOR
C     THE ARRAY HAVE BEEN READ.
C

      SUBROUTINE MINP1(A,IJ,*)
      REAL A(IJ)
      DO 10 J=1,IJ
      A(J)=0.0
   10 CONTINUE
   50 READ(5,1) J,VAL
      IF(J) 100,100,60
   60 A(J)=VAL
      GOTO 50
    1 FORMAT(I10,G14.6)
  100 RETURN 1
      END
```

C SUBROUTINE MINP2

```
C     SUBROUTINE MINP2

C     THIS SUBROUTINE IS USED TO INITIALIZE THE ARRAY A -
C     A REAL ARRAY OF DIMENSION A(IJ,IK)
C
C     FIRST, THE ARRAY IS INITIALIZED TO 0.  THEN THE NON-
C     ZERO ELEMENTS (WITH THEIR INDICES) ARE READ IN AND
C     ENTERED.  AN INDEX OF 0 INDICATES THAT ALL DATA FOR
C     THE ARRAY HAVE BEEN READ.
C

      SUBROUTINE MINP2(A,IJ,IK,*)
      REAL A(IJ,IK)
      DO 10 J=1,IJ
      DO 10 K=1,IK
      A(J,K)=0.
   10 CONTINUE
   50 READ(5,1) J,K,VAL
      IF(J) 100,100,60
   60 A(J,K)=VAL
      GOTO 50
    1 FORMAT(2I10,G14.6)
  100 RETURN 1
      END
```

C   SUBROUTINE MINP3

```
C      SUBROUTINE MINP3

C      THIS SUBROUTINE IS USED TO INITIALIZE THE ARRAY A -
C      A REAL ARRAY OF DIMENSION A(IJ,IK,IL)
C
C      FIRST, THE ARRAY IS INITIALIZED TO 0.  THEN THE NON-
C      ZERO ELEMENTS (WITH THEIR INDICES) ARE READ IN AND
C      ENTERED.  AN INDEX OF 0 INDICATES THAT ALL DATA FOR
C      THE ARRAY HAVE BEEN READ.
C

       SUBROUTINE MINP3(A,IJ,IK,IL,*)
       REAL A(IJ,IK,IL)
     1 FORMAT(3I10,G14.6)
       DO 10 J=1,IJ
       DO 10 K=1,IK
       DO 10 L=1,IL
       A(J,K,L)=0.
    10 CONTINUE
    50 READ(5,1) J,K,L,VAL
       IF(J) 100,100,60
    60 A(J,K,L)=VAL
       GOTO 50
   100 RETURN 1
       END
```

2. TABLES

TABLE C - 1

MODEL DATA REQUIRED BY INPUT

INITIAL SPLIT FACTOR VALUES

1.0			
4			
6			
15			
11			
7			
1			
1			
1	CON		
1	-0.4		
4	0.4		
0	0.0		
10	A		
1	1	2	1.0
2	1	2	0.15
3	1	2	1.00
4	1	2	1.00
2	1	8	0.85
1	2	3	0.92
2	2	3	0.90
3	2	3	0.15
4	2	3	0.98
1	3	4	1.0
2	3	4	0.97
3	3	4	0.90
4	3	4	0.90
1	4	11	0.95
2	4	11	0.95
3	4	11	0.95
4	4	11	0.95
1	5	12	1.00
2	5	12	1.00
3	5	12	1.00
4	5	12	0.05
1	6	7	0.20
2	6	7	0.20
4	6	7	0.95
1	6	14	0.80
2	6	14	0.80
3	6	14	1.00
4	6	14	0.05
1	7	15	1.00
2	7	15	1.00
4	7	15	0.06
2	8	9	0.80

TABLE C - 1 ...CONT'D

2	9	1	0.10
2	9	10	0.90
2	10	1	0.1
1	11	5	0.50
2	11	5	0.50
3	11	5	0.50
4	11	5	0.50
1	11	6	0.50
2	11	6	0.50
3	11	6	0.50
4	11	6	0.50
1	12	3	0.80
2	12	3	0.80
3	12	3	0.80
4	12	3	0.80
1	12	13	0.20
2	12	13	0.20
3	12	13	0.20
4	12	13	0.20
1	13	1	0.90
2	13	1	0.90
3	13	1	0.90
4	13	1	0.90
1	13	2	0.10
2	13	2	0.10
3	13	2	0.10
4	13	2	0.10
1	14	1	0.65
2	14	1	0.65
3	14	1	0.65
4	14	1	0.65
1	14	3	0.35
2	14	3	.35
3	14	3	0.35
4	14	3	0.35
1	15	1	0.05
2	15	1	0.05
3	15	1	0.05
4	15	1	0.05
1	15	3	0.95
2	15	3	0.95
3	15	3	0.95
4	15	3	0.95
00	00	00	0.00
13	Y		.
1	1	1	0.40
2	1	1	0.25
3	1	1	0.35
1	3	2	0.92
2	3	2	0.03

TABLE C - 1 ...CONT'D

3	3	2	0.04
4	3	2	0.01
1	4	3	0.60
3	4	3	0.10
4	4	3	0.30
1	4	4	0.08
2	4	4	0.01
3	4	4	0.01
4	4	4	0.90
1	5	5	0.08
2	5	5	0.01
3	5	5	0.01
4	5	5	0.90
1	6	6	0.30
2	6	6	0.30
4	6	6	0.40
1	7	7	0.10
4	7	7	0.90
2	9	8	-1.00
2	1	9	-1.00
2	10	10	-1.00
2	1	11	-1.00
00	00	00	0.00
14	D		.
1	1	2	0.08
2	1	2	0.10
3	1	2	0.85
4	1	2	0.02
1	2	4	0.05
2	2	4	0.05
3	2	4	0.05
4	2	4	0.05
4	3	5	0.95
3	4	7	1.00
4	4	7	0.95
2	5	8	0.10
2	6	10	0.85
00	00	00	0.00
7	FDCOST	.	
1	1	0.02	
2	1	0.02	
3	1	0.02	
4	1	0.02	
1	2	0.03	
2	2	0.03	
3	2	0.03	
4	2	0.03	
1	3	0.048	
2	3	0.048	
3	3	0.048	

TABLE C - 1 ...CONT'D

4	3	0.048
1	4	0.10
2	4	0.10
3	4	0.10
4	4	0.10
1	5	0.10
2	5	0.10
3	5	0.10
4	5	0.10
1	6	0.038
2	6	0.038
3	6	0.038
4	6	0.038
1	7	0.11
2	7	0.11
3	7	0.11
4	7	0.11
1	8	0.034
2	8	0.034
3	8	0.034
4	8	0.034
1	9	0.084
2	9	0.084
3	9	0.084
4	9	0.084
1	10	0.033
2	10	0.033
3	10	0.033
4	10	0.033
1	11	0.034
2	11	0.034
3	11	0.034
4	11	0.034
00	00	0.000
6	OPCOST	.
2	1	0.032
1	2	0.032
4	4	0.028
4	5	0.0076
4	6	0.0064
4	7	0.0029
2	10	0.052
00	00	0.000
8	VALPRO	.
3	1	0.029
1	2	0.012
2	2	0.012
3	2	0.012
4	2	0.012
4	3	0.12

TABLE C - 1 ...CONT'D

4	4	0.12	
2	5	0.026	
2	6	0.3	
00	00	0.000	
9	CREDVA	.	
1	4	1.0	
1	5	1.00	
4	8	1.0	
5	9	1.0	
6	10	1.0	
7	11	1.0	
00	00	0.000	
12	CPROD		.
3	4	3	1.0
3	4	4	1.0
00	00	00	0.00
11	CINS		.
2	1	2	1.0
2	2	2	1.0
2	3	2	1.0
2	4	2	1.0
4	2	8	-0.8
5	2	9	-0.1
6	2	9	-0.9
7	2	10	-0.05
00	00	00	0.00
15	GBND		.
2	1	1	0.23
1	2	1	0.66
1	3	1	1.15
4	5	1	0.26
4	6	1	0.26
2	10	1	0.14
2	1	2	2.0
1	2	2	2.0
1	3	2	2.0
4	5	2	2.0
4	6	2	2.0
2	10	2	2.0
00	00	00	0.00
16	PBND		.
00	00	00	0.00
5	FBND	.	
1	1	0.96	
1	2	2.0	
2	1	.105	
2	2	2.0	
3	1	0.027	
3	2	2.0	
6	1	0.038	

TABLE C - 1 ...CONT'D

6	2	2.0
7	1	0.045
7	2	2.0
00	00	0.000
2	RHS	
1	0.014	
2	1.05	
3	0.06	
00	0.00	
4	RNGE	
3	0.28	
00	0.00	
3	NCODE	
1	1	
2	1	
3	-1	
4	1	
5	1	
6	1	
7	1	
00	0 00	
00	0 00	

TABLE C-2.

Summary Written by INPUT

```
INPUT OF DATA FOR THE STANDARD L.P. PROBLEM

CATALYST SELECTIVITY =1.00

NO. OF COMPONENTS = 4

NO. OF PRODUCT STREAMS = 6

NO. OF UNITS = 15

NO. OF FRESH FEED STREAMS = 11

NO. OF INEQUALITY CONSTRAINTS =  7

NB= 1     NR= 1

THE ARRAYS READ IN ARE -

ARRAY NAME         CODE NO.
  CON                  1
  A                   10
  Y                   13
  D                   14
  FDCOST               7
  OPCOST               6
  VALPRO               8
  CREDVA               9
  CPROD               12
  CINS                11
  GBND                15
  PBND                16
  FBND                 5
  RHS                  2
  RNGE                 4
  NCODE                3

INPUT FINISHED
```

TABLE C - 3

VARIABLE DEFINITIONS

MODEL DATA READ BY INPUT

```
C                          - DEFINITIONS -
C
C      FX(K)          -KTH EXTERNAL FRESH FEED
C
C      G(J,K)         -TOTAL FEED OF COMPONENT J TO UNIT K
C
C      P(J,K)         -COMPONENT J OF THE KTH EXTERNAL PRODUCT
C
C      X(J)           -THE VARIABLES OF THE L.P. PROBLEM
C
C
C                        - VARIABLE LIST -
C
C      INPUT VARIABLES -
C
C      A(I,J,K)       -FRACTION OF G(I,J) WHICH GOES TO UNIT K
C                     -NAGIEV OR RECOVERY FACTOR
C
C      CINS(I,J,K)    -COEFFICIENT OF G(J,K) IN THE ITH CON-
C                      STRAINT
C
C      CON(I)         -FRACTION OF G(1,3) WHICH WOULD BE CON-
C                      VERTED TO ( OR DISAPPEAR FROM) COMPONENT
C                      I IN UNIT 3 IF THE SELECTIVITY OF THE
C                      CATALYST WERE 1.0
C
C      CPROD(I,J,K)   -COEFFICIENT OF P(J,K) IN THE ITH CON-
C                      STRAINT
C
C      CREDVA(I,K)    -COEFFICIENT OF FX(K) IN THE ITH CON-
C                      STRAINT
C
C      D(I,J,K)       -FRACTION OF G(I,K) WHICH LEAVES THE
C                      PROCESS AS EXTERNAL PRODUCT P(I,J)
C
C      FBND(I,1)      -BOUND ON FX(I)
C      FBND(I,2)      -FLAG INDICATING TYPE OF BOUND
C                     -SAME DEFINITION AS NBT
C
C      FDCOST(I,J)    -COST OF COMPONENT I OF FX(J)
C
C      GBND(I,K,1)    -BOUND ON G(I,K)
C      GBND(I,K,2)    -FLAG INDICATING TYPE OF BOUND
C                     -SAME AS NBT
C
```

TABLE C - 3 ...CONT'D

```
C      NB              -INPUT FLAG INDICATING THE PRESENCE OF A
C                       BOUND VECTOR
C                           NB=0 - NO BOUNDS
C                           NB=1 - BOUNDS REQUIRED
C
C      NBT(K)          -FLAG INDICATING TYPE OF BOUND
C                           0 - NO BOUND ON THIS COLUMN
C                           1 - LOWER BOUND
C                           2 - UPPER BOUND
C                           3 - FIXED VALUE
C                           4 - FREE VARIABLE
C                           5 - LOWER BOUND IS -INFINITY
C                           6 - UPPER BOUND IS +INFINITY
C
C      NCODE(I)        -FLAG INDICATING TYPE OF ITH CONSTRAINT
C                          -1 - GREATER THAN OR EQUAL TO
C                           0 - EQUALITY
C                          +1 - LESS THAN OR EQUAL TO
C
C      NCOMP           -NUMBER OF COMPONENTS
C
C      NCON            -NUMBER OF CONSTRAINTS (EXCEPT MATERIAL
C                       BALANCE CONSTRAINTS)
C
C      NG              -NUMBER OF UNITS - INCLUDING STREAM
C                       SPLITTERS
C
C      NF              -NUMBER OF EXTERNAL FRESH FEEDS
C
C      NPRO            -NUMBER OF EXTERNAL PRODUCT STREAMS
C
C      NR              -INPUT FLAG INDICATING PRESENCE OF RANGE
C                       VECTOR
C                           NR=0 - NO RANGE VECTOR
C                           NR=1 - RANGE VECTOR REQUIRED
C
C      PBND(I,J,1)     -BOUND ON P(I,J)
C      PBND(I,J,2)     -TYPE OF BOUND ON P(I,J)
C                      -SAME AS NBT
C
C      RNGE(I)         -OTHER LIMIT ON RANGE OF RHS(I)
C
C      RHS(I)          -RIGHT HAND SIDE OF CONSTRAINT I
C
C      SEL             -SELECTIVITY OF CATALYST IN UNIT 3
C
C      VALPRO(I,J)     -VALUE OF PRODUCT STREAM P(I,J)
C
C      Y(I,J,K)        -FRACTION OF FEED FX(K), FED TO UNIT K,
C                       WHICH IS COMPONENT I
```

3. DOCUMENTATION

Subroutine INPUT reads in the model data and stores it in the unlabelled COMMON block. The variables and arrays appearing there are defined in table C-3. Subroutine INPUT was developed for model data input to the MPS/360 data generation programs of Appendix B. Advantage was taken of MPS/360 features which allow specification of bounds on variables, and ranges on the requirements, enabling a reduction in the number of constraints. The variable SEL was required in an earlier formulation of the model, and is normally set to 1.0.

The detailed procedure used is documented in the subroutine listings. A listing of the model data required for the butadiene process appears as table C-1. Subroutine INPUT monitors its progress, writing a summary of data read. This summary appears as table C-2.

A better utilization of storage space is possible, but this improvement would require an effort unwarranted for this study.

Subroutine INPUT uses IBM's subroutines MINV and ARRAY (32) for matrix inversion.

APPENDIX D

TWO-PHASE SIMPLEX ALGORITHM

Table of Contents

C SUBROUTINE CANON

```
C     SUBROUTINE CANON
C     PURPOSE
C         TO EXPRESS A GIVEN L.P. PROBLEM IN CANONICAL FORM
C         SUITABLE FOR SOLUTION BY THE TWO-PHASE SIMPLEX
C         ALGORITHM
C
C  ************************************************************
C

      SUBROUTINE CANON
      COMMON /LP/ CON(30,60),RHS(30),CZ(25),VNAME(25,3),
     1X(25),NCODE(30),M,N,NBAS(60),NPH,NS
C     INITIALIZE TABLEAU MATRIX, VECTORS

      M1=M+1
      DO 3 K=1,N
      NBAS(K)=0
      CON(M+2,K)=0.
    3 CONTINUE
      RHS(M+1)=0.
      RHS(M+2)=0.
C     ENTER OBJECTIVE FUNCTION IN CONSTRAINT MATRIX

      DO 4 K=1,N
    4 CON(M+1,K)=CZ(K)
C     MAKE ALL RHS ELEMENTS NON-NEGATIVE

      DO 10 J=1,M
      IF(RHS(J)) 5,10,10
    5 DO 6 K=1,N
    6 CON(J,K)=-CON(J,K)
      RHS(J)=-RHS(J)
      NCODE(J)=-NCODE(J)
   10 CONTINUE
C     IF THIS IS A MAXIMIZATION PROBLEM, CONVERT IT TO MIN.

      IF(NCODE(M+1)) 13,13,11
   11 DO 12 K=1,N
   12 CON(M+1,K)=-CON(M+1,K)
C     ADD SLACK AND SURPLUS VARIABLES.

   13 NO=N
      DO 16 J=1,M
      IF(NCODE(J)) 14,16,15
```

```
      C     SUBROUTINE CANON  ...(CONT'D)
   14 CON(M+2,N+1)=+1.
      NBAS(N+1)=0
      GOTO 151
   15 CON(M+2,N+1)=0.
      NBAS(N+1)=J
  151 N=N+1
      DO 152 K=1,M1
  152 CON(K,N)=0.0
      CON(J,N)=NCODE(J)
   16 CONTINUE
C     ADD ARTIFICIAL VARIABLES AND THE INFEASIBILITY FORM.
      NS=N
      NPH=0
      M2=M+2
      DO 18 J=1,M
      IF(NCODE(J)) 17,17,18
   17 N=N+1
      DO 171 K=1,M2
  171 CON(K,N)=0.
      CON(J,N)=1.0
      NBAS(N)=J
      RHS(M+2)=RHS(M+2)-RHS(J)
      DO 18 K=1,NO
      CON(M+2,K)=CON(M+2,K)-CON(J,K)
   18 CONTINUE
C     VARIABLES
C     STORE FLAG INDICATING PHASE, STORE NO. OF LEGITIMATE
      NCODE(M+2)=NO
      IF(N-NS) 19,19,20
   19 NPH=1
   20 CONTINUE
      RETURN
      END
```

```
        C    SUBROUTINE TPHSIM

C     SUBROUTINE TPHSIM

C     PURPOSE
C          TO SOLVE AN L.P. PROBLEM, GIVEN IN THE APPROPRIATE
C          CANONICAL FORM, USING THE TWO PHASE SIMPLEX
C          ALGORITHM
C
C   ************************************************************
C

      SUBROUTINE TPHSIM
      COMMON /LP/ CON(30,60),RHS(30),CZ(25),VNAME(25,3),
     1X(25),NCODE(30),M,N,NBAS(60),NPH,NS
      MO=M+2-NPH

C     FIND THE MINIMUM COST CON(MO,J) AND HENCE THE VARI-
C     ABLE TO ENTER THE BASIS.

    1 VMIN =-1.0E-06
      NIN=0
      DO 10 J=1,N
      IF(CON(MO,J)-VMIN) 5,10,10
    5 VMIN=CON(MO,J)
      NIN=J
   10 CONTINUE

C     TEST FOR A MINIMUM.

      IF(NIN) 24,24,11

C     CHOOSE THE VARIABLE TO LEAVE THE BASIS.

   11 NOUT=0
      VOUT=1.0E+30
      DO 14 J=1,M
      IF(CON(J,NIN)) 14,14,12
   12 V=RHS(J)/CON(J,NIN)
      IF(V-VOUT) 13,14,14
   13 VOUT=V
      NOUT=J
   14 CONTINUE

C     CHECK FOR AN UNBOUNDED SOLUTION.

      IF(NOUT) 32,32,15

C     PERFORM THE PIVOT OPERATIONS NECESSARY TO UPDATE THE
C     PROBLEM.

   15  DO 17 K=1,N
```

```
      C     SUBROUTINE TPHSIM  ...(CONT'D)
      IF(NBAS(K)-NOUT) 17,16,17
   16  NBAS(K)=0
      GOTO 18
   17  CONTINUE
   18 NBAS(NIN)=NOUT
      DO 22 K=1,N
      IF(NBAS(K)) 19,19,22
   19 CON(NOUT,K)=CON(NOUT,K)/CON(NOUT,NIN)
      DO 21 J=1,MO
      IF(J-NOUT) 20,21,20
   20 CON(J,K)=CON(J,K)-CON(NOUT,K)*CON(J,NIN)
   21  CONTINUE
   22 CONTINUE
      RHS(NOUT)=RHS(NOUT)/CON(NOUT,NIN)
      DO 23 J=1,MO
      IF(J-NOUT) 42,23,42
   42 RHS(J)=RHS(J)-RHS(NOUT)*CON(J,NIN)
   23  CON(J,NIN)=0.
      CON(NOUT,NIN)=1.
      GOTO 1
C     IS THE MINIMUM PHASE 1 OR PHASE 2
   24 IF(NPH) 25,25,30
C     PHASE 1 IS COMPLETE.  IS THE SOLUTION FEASIBLE
   25 IF(RHS(M+2)+1.E-06) 31,26,26
C     THE SOLUTION IS FEASIBLE.  PREPARE FOR PHASE 2.
   26 MO=MO-1
      N=NS
      NPH=1
      GOTO 1
C     SET FLAG FOR OPTIMAL SOLUTION
   30 NS=0
      RHS(M+1)=-RHS(M+1)
      RETURN
C     SET FLAG FOR NO FEASIBLE SOLUTION.
   31 NS=1
      RETURN
C     SET FLAG FOR UNBOUNDED SOLUTION.
   32 NS=2
```

```
C     SUBROUTINE TPHSIM  ...(CONT'D)
      RETURN
      END
```

```
      C     SUBROUTINE LPSOL

C     SUBROUTINE LPSOL

C     PURPOSE
C         TO INTERPRET THE SOLUTION TABLEAU, PRINT ERROR
C         MESSAGES IF NECESSARY, AND PRINT THE OPTIMAL
C         SOLUTION IF DESIRED
C
C   ************************************************************
C

      SUBROUTINE LPSOL(KK)
      COMMON /LP/ CON(30,60),RHS(30),CZ(25),VNAME(25,3),
     1X(25),NCODE(30),M,N,NBAS(60),NPH,NS
  100 FORMAT(/////,12X,'NO FEASIBLE SOLUTION')
  101 FORMAT(/////,12X,'UNBOUNDED SOLUTION')
  103 FORMAT('0',12X,'OBJECTIVE FUNCTION =',G12.4,/)
  104 FORMAT('0',12X,'VARIABLE NAME',10X,'VALUE',/)
  105 FORMAT(16X,3A4,F14.5)

C     PRINT ERROR MESSAGES IF NECESSARY

      IF(NS-1) 20,10,15
   10 WRITE(6,100)
      RETURN
   15 WRITE(6,101)
      RETURN

C     LOCATE AND STORE OBJECTIVE FUNCTION AND DECISION
C     VARIABLE VALUES

   20 NO=NCODE(M+2)
      DO 23 J=1,NO
      IF(NBAS(J)) 21,21,22
   21 X(J)=0.0
      GOTO 23
   22 X(J)=RHS(NBAS(J))
   23 CONTINUE
      IF(NCODE(M+1)) 25,25,24
   24 X(NO+1)=-RHS(M+1)
      GOTO 30
   25 X(NO+1)=RHS(M+1)

C     PRINT THE OPTIMAL SOLUTION IF DESIRED (KK.NE.0)

   30 IF(KK.EQ.0) RETURN
      WRITE(6,103) X(NO+1)
      WRITE(6,104)
      DO 35 J=1,NO
   35 WRITE(6,105) (VNAME(J,K),K=1,3),X(J)
      RETURN
```

```
C       SUBROUTINE LPSOL   ...(CONT'D)
      END
```

2. TABLES

TABLE D - 1

SAMPLE - INPUT VARIABLE NAMES
FOR LPSOL

FX 1
FX 2
FX 3
FX 4
FX 5
FX 6
FX 7
FX 8
FX 9
FX10
FX11

3. DOCUMENTATION

3.1 Definition of Variables

Communication between CANON, TPHSIM, LPSOL and the calling program is via the COMMON block /LP/. The variables located there are:

CON(I,J)	-	constraint coefficient matrix (destroyed)
RHS(I)	-	constraint requirement vector (destroyed)
CZ(J)	-	objective function coefficients
VNAME(J,.)	-	variable names for output via LPSOL
X(J)	-	optimal solution values of variables x(N+1) = optimal objective function value
NCODE(I)	-	code vector indicating constraint type NCODE(I) = +1 - $\geq$ NCODE(I) = 0 - = NCODE(I) = -1 - $\leq$ NCODE(M+1) = problem type, <0 - min
M	-	no. of constraints
N	-	no. of variables (destroyed)
NBAS(J)	-	internal code indicating locational variables in the basis. For NBAS(J)=I,

		I >0, the j^{th} variable is in the basis at level RHS(I)
NPH	-	internal code indicating phase no.
NS	-	on exit, solution status code. NS = 0 - If an optimal solution was reached.

These are the definitions of variables on entry to or (where appropriate) exit from the simplex algorithm subroutines. Those variables destroyed during computation are so indicated.

3.2 Input - Output

Subroutine CANNON requires that the optimization problem be specified via the objective function coefficient vector CZ(J), the constraint coefficient matrix CON(I,J), the requirements vector RHS(I), the constraint and the problem type code vector NCODE(I), the no. of constraints AA, and no. of variables N. The optimal solution variable and objective function values are placed in X(J) and written on logical unit 6 if desired (LPSOL input parameter kk = 0). If an optimal solution was not reached, an appropriate error message is written. A sample of the card input of variable names (VNAME(J,.)) required by LPSOL appears in table B-1.

3.3 Procedure

The procedure followed is that suggested by Dantzig (25). Subroutine CANON prepares the tableau for the simplex method, subroutine TPHSIM solves the L.P. problem using the two-phase simplex method and LPSOL handles the solution output.

APPENDIX E

OPTIMIZATION MODEL GENERATION

Table of Contents

C SUBROUTINE INIT

```
C     SUBROUTINE INIT

C     PURPOSE
C         MODIFICATION OF INPUT MODEL DATA TO ACCOUNT FOR
C         RANGES SPECIFIED ON COSTRAINT REQUIREMENTS VECTOR
C
C  ************************************************************

      SUBROUTINE INIT


      COMMON A(4,15,15),Y(4,15,11),D(4,6,15),CPROD(10,4,6),
     1CINS(10,4,15),CREDVA(10,11),FDCOST(4,11),
     1OPCOST(4,15),VALPRO(4,6),CON(4),RHS(10),RNGE(10),
     1GBND(4,15,2),PBND(4,6,2),FBND(11,2),SEL,NCODE(10),
     1NCOMP,NPRO,NG,NF,NCON,NB,NR
      COMMON/LP/ C(30,60),RH(30),CZ(25),VNAME(25,3),
     1X(25),NC(30),M,N,NBAS(60),NPH,NS

C  IF THERE IS NO RANGE DATA, RETURN

      IF(NR) 11,11,4
      M=NCON

C  ADDITION OF CONSTRAINT TO ACCOUNT FOR EACH RANGE ON A
C  REQUIREMENT

    4 DO 10 J=1,NCON
      IF(RNGE(J)-0.0001) 10,10,5
    5 M=M+1

C  NEW REQUIREMENT, CONSTRAINT TYPE CODE

      NCODE(M)=-NCODE(J)
      RHS(M)=RHS(J)+RNGE(J)*NCODE(M)

C  NEW CONSTRAINT COEFFICIENTS

      DO 6 K=1,NF
    6 CREDVA(M,K)=CREDVA(J,K)
      DO 7 K=1,NCOMP
      DO 8 L=1,NG
    8 CINS(M,K,L)=CINS(J,K,L)
      DO 9 L=1,NPRO
    9 CPROD(M,K,L)=CPROD(J,K,L)
    7 CONTINUE
   10 CONTINUE

C  UPDATE CONSTRAINT TOTAL
```

```
C      SUBROUTINE INIT   ...(CONT'D)

      NCON=M
   11 RETURN
      END
```

C SUBROUTINE COEFF

```
C     SUBROUTINE COEFF
C
C     PURPOSE
C        TO CALCULATE THE COEFFICIENT MATRICES FOR THE
C        SYSTEM MODEL TRANSFORMATION EQUATIONS WHICH
C        EXPRESS INTERNAL AND PRODUCT STREAM RATES IN
C        TERMS OF EXTERNAL FEED RATES
C
C     SUBROUTINES REQUIRED
C        ARRAY, MINV
C
C  ******************************************************************
C

      SUBROUTINE COEFF(BIY,DBIY)


      REAL BIY(4,15,11),DBIY(4,6,11)
      DIMENSION B(15,15),WV(6),IW1(15),IW2(15)
      COMMON A(4,15,15),Y(4,15,11),D(4,6,15),CPROD(10,4,6),
     1CINS(10,4,15),CREDVA(10,11),FDCOST(4,11),
     1OPCOST(4,15),VALPRO(4,6),CON(4),RHS(10),RNGE(10),
     1GBND(4,15,2),PBND(4,6,2),FBND(11,2),SEL,NCODE(10),
     1NCOMP,NPRO,NG,NF,NCON,NB,NR

C  GENERATE COEFFICIENT MATRIX FOR INTERNAL STREAM EQUATION

      IDB=15
    4 DO 100 I=1,NCOMP

C  SET UP ITH DIAGONAL BLOCK OF PARTITIONED B MATRIX

      DO 10 J=1,NG
      DO 10 K=1,NG
      B(J,K)=-A(I,K,J)
      IF(J-K) 10,5,10
    5 B(J,K)=B(J,K)+1.
   10 CONTINUE
      IF(I.EQ.1) B(4,3)=B(4,3)-A(1,3,4)*CON(1)

C  INVERT MATRIX

      MODE=2
      CALL ARRAY(MODE,NG,NG,IDB,IDB,B,B)
      CALL MINV(B,NG,DET,IW1,IW2)
      IF(DET) 15,11,15
   11 WRITE(6,1000)
 1000 FORMAT(1H0,12X,'NO INVERSE')
      STOP
   15 MODE=1
```

```
      C      SUBROUTINE COEFF   ...(CONT'D)
      CALL ARRAY(MODE,NG,NG,IDB,IDB,B,B)
C   CALCULATE COEFFICIENTS
      DO 20 J=1,NG
      DO 20 K=1,NF
      BIY(I,J,K)=0.
      DO 19 L=1,NG
   19 BIY(I,J,K)=BIY(I,J,K)+B(J,L)*Y(I,L,K)
   20 CONTINUE
C   INSERT REACTION CONVERSION TERMS
      IF(CON(I)) 100,100,25
   25 CS=A(I,3,4)*CON(4)
      DO 30 J=1,NF
      CSP=CS*BIY(1,3,J)
      DO 30 K=1,NG
      BIY(I,K,J)=BIY(I,K,J)+B(K,4)*CSP
   30 CONTINUE
  100 CONTINUE
C   GENERATE COEFFICIENT MATRIX FOR PRODUCT STREAM EQUATION
  280 DO 300 I=1,NCOMP
      DO 300 J=1,NF
      DO 290 K=1,NPRO
      WV(K)=0.
      DO 290 L=1,NG
      WV(K)=WV(K)+D(I,K,L)*BIY(I,L,J)
  290 CONTINUE
      DO 300 K=1,NPRO
      DBIY(I,K,J)=WV(K)
  300 CONTINUE
      RETURN
      END
```

C SUBROUTINE CONT2

```
C   SUBROUTINE CONT2

C      PURPOSE
C          TO SET UP THE REDUCED FORM OF THE LINEAR
C          OPTIMIZATION MODEL, GIVEN MODEL DATA AND THE
C          APPROPRIATE TRANSFORMATION EQUATION COEFFICIENTS.
C          DEPENDING ON IRCODE, GENERATION OF OBJECTIVE
C          FUNCTION COEFFICIENTS CAN BE AVOIDED
C
C   ***************************************************************

      SUBROUTINE CONT2(IRCODE,BIY,DBIY)


      REAL BIY(4,15,11),DBIY(4,6,11)
      COMMON A(4,15,15),Y(4,15,11),D(4,6,15),CPROD(10,4,6),
     1CINS(10,4,15),CREDVA(10,11),FDCOST(4,11),
     1OPCOST(4,15),VALPRO(4,6),CON(4),RHS(10),RNGE(10),
     1GBND(4,15,2),PBND(4,6,2),FBND(11,2),SEL,NCODE(10),
     1NCOMP,NPRO,NG,NF,NCON,NB,NR
      COMMON/LP/ C(30,60),RH(30),CZ(25),VNAME(25,3),
     1X(25),NC(30),M,N,NBAS(60),NPH,NS

C   OMIT OBJECTIVE FUNCTION COEFFICIENTS IF DESIRED

      IF(IRCODE-1) 100,100,202

C   OBJECTIVE FUNCTION

  100 DO 400 J=1,NF
      CZ(J)=0.
      DO 400 K=1,NCOMP
      SUM=0.
      DO 399 L=1,NG
  399 SUM=SUM+Y(K,L,J)
      CZ(J)=CZ(J)+SUM*FDCOST(K,J)
  400 CONTINUE
  190 DO 200 I=1,NCOMP
      DO 200 J=1,NG
      DO 200 K=1,NF
      CZ(K)=CZ(K)+OPCOST(I,J)*BIY(I,J,K)
  200 CONTINUE
      DO 310 I=1,NCOMP
      DO 310 J=1,NPRO
      DO 310 K=1,NF
      CZ(K)=CZ(K)-VALPRO(I,J)*DBIY(I,J,K)
  310 CONTINUE

C   CONSTRAINTS
```

```
      C   SUBROUTINE CONT2      ...(CONT'D)
  202 M=NCON
C   REQUIREMENTS VECTOR, CONSTRAINT TYPE CODE VECTER
  405 DO 410 J=1,NCON
      RH(J)=RHS(J)
      NC(J)=NCODE(J)
C   FEED STREAM CONSTRAINT COEFFICIENTS
      DO 410 K=1,NF
  410 C(J,K)=CREDVA(J,K)
C   FEED STREAM BOUNDS - CONVERT TO CONSTRAINTS
      IF(NB) 204,204,411
  411 DO 440 J=1,NF
      IF(FBND(J,2)-0.01) 440,440,415
  415 M=M+1
      DO 430 K=1,NF
  430 C(M,K)=0.
      C(M,J)=1.
      RH(M)=FBND(J,1)
      NC(M)=(3-FBND(J,2))*(3*FBND(J,2)-4)/2.
  440 CONTINUE
C   INTERNAL STREAM CONSTRAINT COEFFICIENTS
  204 DO 210 I=1,NCON
      DO 210 J=1,NCOMP
      DO 210 K=1,NG
      IF(CINS(I,J,K)) 205,210,205
  205 DO 209 L=1,NF
  209 C(I,L)=C(I,L)+CINS(I,J,K)*BIY(J,K,L)
  210 CONTINUE
C   INTERNAL STREAM BOUNDS - CONVERT TO CONSTRAINTS
      IF(NB) 311,311,211
  211 DO 230 I=1,NCOMP
      DO 230 J=1,NG
      IF(GBND(I,J,2)-0.01) 230,230,215
  215 M=M+1
      DO 220 K=1,NF
      C(M,K)=BIY(I,J,K)
  220 CONTINUE
      RH(M)=GBND(I,J,1)
      NC(M)=(3-GBND(I,J,2))*(3*GBND(I,J,2)-4)/2
  230 CONTINUE
```

```
      C   SUBROUTINE CONT2       ...(CONT'D)
C   PRODUCT STREAM CONSTRAINT COEFFICIENTS

  311 DO 320 I=1,NCON
      DO 320 J=1,NCOMP
      DO 320 K=1,NPRO
      IF(CPROD(I,J,K)) 315,320,315
  315 DO 319 L=1,NF
  319 C(I,L)=C(I,L)+CPROD(I,J,K)*DBIY(J,K,L)
  320 CONTINUE

C   PRODUCT STREAM BOUNDS - CONVERT TO CONSTRAINTS

      IF(NB) 350,350,321
  321 DO 340 I=1,NCOMP
      DO 340 J=1,NPRO
      IF(PBND(I,J,2)-0.01) 340,340,325
  325 M=M+1
       DO 330 K=1,NF
  330 C(M,K)=DBIY(I,J,K)
      RH(M)=PBND(I,J,1)
      NC(M)=(3-PBND(I,J,2))*(3*PBND(I,J,2)-4)/2
  340 CONTINUE

C   DEFINE NO. OF VARIABLES, PROBLEM TYPE CODE

  350 N=NF
      NC(M+1)=-1
      RETURN
      END
```

2. DOCUMENTATION

Subroutines INIT, COEFF and CONT2 generates the standard form of the reduced optimization model and specifies the entries in the COMMON block /LP/, required for solution by the two-phase simplex algorithm (Appendix D).

Subroutine INIT modifies the model data read by INPUT by converting any ranges specified on the requirements vector to additional constraints.

Subroutine COEFF generates the coefficients of the required transformation equations:

$$\underline{g} = \underline{B}^{-1}\ \underline{fx}\ \underline{Y}^{*} \tag{B-1}$$

$$\underline{pr} = \underline{T}^{*}\underline{B}^{-1}\ \underline{fx}\ \underline{Y}^{*} \tag{B-2}$$

and stores them in arrays BIY and DBIY. The matrix inverse $\underline{B}^{-1}$ is required. So doing for the inverse by partitioning the matrix enables more efficient calculation of the inverse since $\underline{B}$ is sparse, and, for this process, block lower triangular.

Subroutine CONT2 generates the objective function and constraint matrix for the reduced problem, converting bounds into additional constraints, and stores the required information in the COMMON block /LP/.

APPENDIX F

DETERMINISTIC DECISION, PATTERN SEARCH

Table of Contents

```
        C     MAINLINE -- D.P.

C     MAINLINE -- D.P.  DETERMINISTIC PROBLEM

C     PURPOSE
C         TO FIND A SOLUTION TO THE DETERMINISTIC
C         OPTIMIZATION MODEL, BUTADIENE AREA
C
C     SUBROUTINES REQUIRED
C
C         INPUT - IMINP1, MINP1, MINP2,MINP3
C         INIT
C         VAL - COEFF, CONT2, CANON, TPHSIM, LPSOL, STPRNT
C         BPSH - VAL (DUMMY NAME FCT), BPSOUT
C
C  ***********************************************************
C
C  SPECIFICATION OF VARIABLES REQUIRED FOR BPSH

      EXTERNAL VAL
      REAL B(5),T(5),DX(5),DXM(5),BND(2,5)

C  COMMON SPECIFICATION, MODEL DATA

      COMMON A(4,15,15),Y(4,15,11),D(4,6,15),CPROD(10,4,6),
     1CINS(10,4,15),CREDVA(10,11),FDCOST(4,11),
     1OPCOST(4,15),VALPRO(4,6),CON(4),RHS(10),RNGE(10),
     1GBND(4,15,2),PBND(4,6,2),FBND(11,2),SEL,NCODE(10),
     1NCOMP,NPRO,NG,NF,NCON,NB,NR

C  COMMON SPECIFICATIONS, REQUIRED BY SUBROUTINE VAL

      COMMON /V/ IV(5,3),NFIN
      COMMON /LP/ CM(30,60),RH(30),CZ(25),VNAME(25,3),
     1X(25),NC(30),M,NCOL,NBAS(60),NPH,NS

C  INPUT - OUTPUT FORMATS

    1 FORMAT(3I10)
    2 FORMAT(5G15.5)
    3 FORMAT('1'////13X,'THIS IS THE OPTIMAL SOLUTION')
    4 FORMAT('1FINISHED')
    5 FORMAT(3A4)
  105 FORMAT('1'////13X,'PATTERN SEARCH SOLUTION'//15X
     *,'VARIABLE',
     15X,'INITIAL',5X,'MINIMUM',6X,'UPPER',7X,'LOWER'/27X
     *,'STEP SIZE'
     2,3X,'STEP SIZE',5X,'BOUND',7X,'BOUND'/)
  106 FORMAT(13X,'A(I,',I2,',',I2,') ',2G12.4,F9.2,F12.2)
  107 FORMAT('0',12X,'MAXIMUM NO. OF CYCLES =',I5//13X
     *,'EPSILON =',
     1G10.3)
```

```
      C     MAINLINE -- D.P.  ...(CONT'D)
  108 FORMAT('0',12X,'INITIAL SOLUTION')
C   INPUT OF DATA FOR PATTERN SEARCH, INITIALIZATION
      READ(5,1) N,MIT
      DO 10 J=1,N
   10 READ(5,1) (IV(J,K),K=1,3)
      READ(5,2)(B(J),J=1,N)
      READ(5,2) (DX(J),J=1,N)
      READ(5,2) (DXM(J),J=1,N)
      DO 12 K=1,2
   12 READ(5,2) (BND(K,J),J=1,N)
      READ(5,2) EPS
   13 CONTINUE
C   INPUT OF VARIABLE NAMES FOR L.P. OUTPUT
      DO 20 J=1,21
   20 READ(5,5) (VNAME(J,K),K=1,3)
C   INPUT, INITIALIZATION OF MODEL DATA
      CALL INPUT
      CALL INIT
C   OUTPUT OF INITIAL CONDITIONS
      NFIN=2
      WRITE(6,105)
      WRITE(6,106)((IV(J,K),K=1,2),DX(J),DXM(J),(BND(L,J)
     *,L=1,2),J=1,N)
      WRITE(6,107) MIT,EPS
      WRITE(6,108)
      CALL VAL(F,B,N)
C   PATTERN SEARCH SOLUTION
      NFIN=-1
      CALL BPSH(VAL,B,DX,DXM,T,BND,EPS,N,MIT,IER)
      IF(IER.EQ.-1) STOP
C   OUTPUT OF OPTIMAL SOLUTION
      NFIN=1
      WRITE(6,3)
      CALL VAL(F,T,N)
      WRITE(6,4)
      STOP
      END
```

```
          C      SUBROUTINE BPSH

C       SUBROUTINE BPSH

C       PURPOSE
C           TO FIND A LOCAL MINIMUM OF A FUNCTION OF SEVERAL
C           VARIABLES BY THE METHOD OF HOOKE AND JEEVES
C
C       USAGE
C           CALL BPSH(FCT,B,DX,DXM,T,BND,EPS,N,MIT,IER)
C           PARAMETER FCT REQUIRES AN EXTERNAL STETEMENT
C
C       SUBROUTINES REQUIRED
C           FCT,BPSOUT
C
C    **************************************************************
C

        SUBROUTINE BPSH(FCT,B,DX,DXM,T,BND,EPS,N,MIT,IER)
        REAL B(5),T(5),DX(5),DXM(5),BND(2,5)

C    INITIALIZE COUNTERS AND FLAGS

        IER=0
        KK=1
        K=0
        IT=0

C    INITIALIZE TEMPORARY HEAD VECTOR

        CALL FCT(FX,B,N)
        FT=FX
        DO 10 J=1,N
     10 T(J)=B(J)

C    START ITERATION LOOP
C
C    PRINT PATTERN STATUS IF DESIRED

     15 CALL BPSOUT(FT,T,N,IT,KK,K,IER)
        IT=IT+1

C    TEST ITERATION COUNTER

        IF(IT.GT.MIT) GOTO 90

C    PATTERN PERTURBATION TO IMPROVE FUNCTION VALUE

        DO 50 J=1,N

C    POSITIVE PERTURBATION
```

```
      C     SUBROUTINE BPSH   ...(CONT'D)
    TT=T(J)
    IF(TT.GE.BND(1,J)-EPS) GOTO 20
    D=T(J)+DX(J)
    IF(D.LE.BND(1,J)) GOTO 18
    D=BND(1,J)
 18 T(J)=D
    CALL FCT(FN,T,N)
    IF(FN.LE.FT-EPS) GOTO 40
    T(J)=TT
C   NEGATIVE PERTURBATION
    IF(TT.LE.BND(2,J)+EPS) GOTO 50
 20 D=T(J)-DX(J)
    IF(D.GE.BND(2,J)) GOTO 30
    D=BND(2,J)
 30 T(J)=D
    CALL FCT(FN,T,N)
    IF(FN.LE.FT-EPS) GOTO 40
    T(J)=TT
    GOTO 50
 40 FT=FN
 50 CONTINUE
C   TEST FOR FUNCTION VALUE IMPROVEMENT
    IF(FT.GT.FX-EPS) GOTO (82,70),KK
C   EXTEND PATTERN,DEFINE NEW BASE POINT AND INITIAL
C   TEMPORARY HEAD
    KK=1
    K=0
    DO 60 J=1,N
    D=T(J)
    T(J)=2.0*T(J)-B(J)
    IF(T(J).GT.BND(1,J))  T(J)=BND(1,J)
    IF(T(J).LT.BND(2,J))  T(J)=BND(2,J)
    B(J)=D
 60 CONTINUE
    FX=FT
    CALL FCT(FT,T,N)
C   END ITERATION LOOP
    GOTO 15
C   NO IMPROVEMENT OVER BASE POINT, DECREASE STEP SIZE
 70 K=0
```

```
      C     SUBROUTINE BPSH   ...(CONT'D)
      DO 80 J=1,N
      IF(DX(J).LE.DXM(J)+EPS) GOTO 80
      K=1
      DX(J)=DX(J)/2.
      IF(DX(J).LT.DXM(J)) DX(J)=DXM(J)
   80 CONTINUE
C  IF REQUIRED, BEGIN NEW CYCLE
      IF(K.GT.0) GOTO 15
C  STEPSIZE AT MINIMUM, SEARCH CONVERGED
      IER=IT
C  PRINT SOLUTION STATUS IF DESIRED, END SEARCH
      CALL BPSOUT(FT,T,N,IT,KK,K,IER)
      B(1)=FT
      RETURN
C  NO IMPROVEMENT OVER INITIAL TEMPORARY HEAD,PATTERN
C  DESTROYED, RETREAT TO BASE POINT
   82 DO 85 J=1,N
   85 T(J)=B(J)
      KK=2
      FT=FX
C  BEGIN NEW CYCLE
      GOTO 15
C  NO CONVERGENCE IN MIT ITERATIONS, PRINT SOLUTION STATUS,
C  END SEARCH
   90 IER=-1
      CALL BPSOUT(FT,T,N,IT,KK,K,IER)
      B(1)=FT
      RETURN
      END
```

C SUBROUTINE BPSOUT

```
C     SUBROUTINE BPSOUT
C
C     PURPOSE
C           TO PRINT STATUS OF PATTERN SEARCH FOR OPTIMAL
C           SPLIT FACTORS, PRIOR TO EACH NEW CYCLE AND
C           AFTER COMPLETION
C
C   ******************************************************************
C

      SUBROUTINE BPSOUT(F,B,N,IT,KK,K,IER)
      REAL B(1)
  100 FORMAT('1',12X,'PATTERN SEARCH FOR OPTIMAL SPLIT
     * FACTORS'//)
  101 FORMAT(12X,'CYCLE NO. =',I5)
  102 FORMAT(12X,'CONTINUATION NO IMPROVEMENT, RETREAT, '
     1,'PATTERN DESTROYED')
  103 FORMAT(12X,'HALVE STEP SIZE')
  104 FORMAT(17X,'FUNCTION VALUE =',G13.5/17X,
     1'SPLIT FACTORS  =',5(F10.5,/33X))
  105 FORMAT(12X,'TOO MANY ITERATIONS, SOLUTION TERMINATED')
  106 FORMAT(12X,'SEARCH COMPLETED')

C   PRINT TITLE

      IF(IT.EQ.0) WRITE(6,100)

C   PRINT CYCLE NO.

      WRITE(6,101) IT

C   IF NO IMPROVEMENT RESULTED, PRINT ACTION TAKEN

      IF(KK+K.EQ.2) WRITE(6,102)
      IF(K.EQ.1) WRITE(6,103)

C   PRINT FUNCTION VALUE, SPLIT FACTORS -- NEW INITIAL
C   TEMPORARY HEAD

      WRITE(6,104) F,(B(J),J=1,N)

C   IF SEARCH TERMINATED, PRINT MESSAGE

      IF(IER.EQ.-1) WRITE(6,105)
      IF(IER.GT.0) WRITE(6,106)
      RETURN
      END
```

```
          C     SUBROUTINE VAL

C     SUBROUTINE VAL

C     PURPOSE
C         TO EVALUATE THE OBJECTIVE FUNCTION FOR A SPECIFIED
C         TO EVALUATE THE OBJECTIVE FUNCTION FOR A
C         SPECIFIED SET OF SPLIT FACTORS
C
C     SUBROUTINES REQUIRED
C         COEFF, CONT2, CANON, TPHSIM, LPSOL, STPRNT
C
C  ************************************************************
C

      SUBROUTINE VAL(F,S,NN)
      REAL S(5)
      REAL BIY(4,15,11),DBIY(4,6,11)
      COMMON A(4,15,15),Y(4,15,11),D(4,6,15),CPROD(10,4,6),
     1CINS(10,4,15),CREDVA(10,11),FDCOST(4,11),
     1OPCOST(4,15),VALPRO(4,6),CON(4),RHS(10),RNGE(10),
     1GBND(4,15,2),PBND(4,6,2),FBND(11,2),SEL,NCODE(10),
     1NCOMP,NPRO,NG,NF,NCON,NB,NR
      COMMON /V/ IV(5,3),NFIN
      COMMON/LP/ C(30,60),RH(30),CZ(25),VNAME(25,3),
     1X(25),NC(30),M,N,NBAS(60),NPH,NS
  100 FORMAT(/(15X,'A(I,',I2,',',I2,') ',F16.5))

C  INSERTION OF SPLIT FACTORS INTO MODEL DATA

    5 DO 10 J=1,NN
      DO 10 K=1,NCOMP
      A(K,IV(J,1),IV(J,2))=S(J)
      A(K,IV(J,1),IV(J,3))=1.0-S(J)
   10 CONTINUE
   11 CONTINUE

C  GENERATION OF MODEL DATA FOR THE REDUCED FORM OF
C  THE L.P. PROBLEM

      CALL COEFF(BIY,DBIY)
      CALL CONT2(1,BIY,DBIY)

C  GENERATION OF L.P. PROBLEM AND CANONICAL FORM, AND
C  SOLUTION BY THE TWO PHASE SIMPLEX ALGORITHM

      CALL CANON
      CALL TPHSIM
      F=RH(M+1)

C  OUTPUT OF RESULTS ON PRINTER IF DESIRED
```

```
      C     SUBROUTINE VAL     ...(CONT'D)

      IF(NFIN) 20,30,40
   20 CALL LPSOL(0)
   30 RETURN
   40 CALL LPSOL(1)
      WRITE(6,100)((IV(J,K),K=1,2),S(J),J=1,NN)
      IF(NFIN.EQ.1) CALL STPRNT(BIY,DBIY)
      RETURN
      END
```

C SUBROUTINE STPRNT

```
C     SUBROUTINE STPRNT

C     PURPOSE
C          TO EVALUATE AND PRINT INTERNAL AND PRODUCT
C          STREAM FLOW RATES, AND TRANSFORMATION EQUATION
C          COEFFICIENTS
C
C   ***********************************************************
C

      SUBROUTINE STPRNT(BIY,DBIY)
      REAL BIY(4,15,11),DBIY(4,6,11),G(4,15),P(4,6)
      COMMON/LP/ C(30,60),RH(30),CZ(25),VNAME(25,3),
     1X(25),NC(30),M,N,NBAS(60),NPH,NS
  102 FORMAT(//13X,'INTERNAL STREAM - G(I,J)'/
     116X,'J',3X,'I = ',4(I5,5X)/)
  103 FORMAT(//13X,'PRODUCT STREAMS - P(I,J)'/
     116X,'J',3X,'I = ',4(I5,5X)/)
  104 FORMAT(12X,I5,2X,4F10.5)

C  EVALUATE INTERNAL AND PRODUCT STREAMS

      DO 50 I=1,4
      DO 45 J=1,15
      G(I,J)=0.0
      DO 45 K=1,11
   45 G(I,J)=G(I,J)+BIY(I,J,K)*X(K)
      DO 50 J=1,6
      P(I,J)=0.0
      DO 50 K=1,11
   50 P(I,J)=P(I,J)+DBIY(I,J,K)*X(K)

C  PRINT RESULTS

      WRITE(6,102) (I,I=1,4)
      DO 60 J=1,15
   60 WRITE(6,104) J,(G(I,J),I=1,4)
      WRITE(6,103) (I,I=1,4)
      DO 70 J=1,6
   70 WRITE(6,104) J,(P(I,J),I=1,4)
      RETURN
      END
```

2. TABLES

TABLE F - 1

INPUT DATA FOR PATTERN SEARCH INITIALIZATION

5	50			
11	5	6		
12	3	13		
13	1	2		
14	1	3		
15	1	3		
0.5	0.8	0.9	0.65	0.05
0.064	0.064	0.064	0.064	0.064
0.001	0.001	0.001	0.001	0.001
1.00	1.0	1.0	1.0	1.0
0.00	0.0	0.0	0.0	0.0
1.0 E-06				

TABLE F-2.

Initial Conditions, Intermediate Results

PATTERN SEARCH SOLUTION

VARIABLE	INITIAL STEP SIZE	MINIMUM STEP SIZE	UPPER BOUND	LOWER BOUND
A(I,11, 5)	0.6400E-01	0.1000E-02	1.00	0.0
A(I,12, 3)	0.6400E-01	0.1000E-02	1.00	0.0
A(I,13, 1)	0.6400E-01	0.1000E-02	1.00	0.0
A(I,14, 1)	0.6400E-01	0.1000E-02	1.00	0.0
A(I,15, 1)	0.6400E-01	0.1000E-02	1.00	0.0

MAXIMUM NO. OF CYCLES = 50

EPSILON = 0.100E-05

INITIAL SOLUTION

OBJECTIVE FUNCTION = -0.2086E-01

VARIABLE NAME	VALUE
FX 1	0.73561
FX 2	0.10500
FX 3	0.0
FX 4	0.0
FX 5	0.01400
FX 6	0.03800
FX 7	0.0
FX 8	0.0
FX 9	0.01556
FX 10	0.0
FX 11	0.00700
A(I,11, 5)	0.50000
A(I,12, 3)	0.80000
A(I,13, 1)	0.90000
A(I,14, 1)	0.65000
A(I,15, 1)	0.05000

TABLE F-2 Continued

```
 PATTERN SEARCH FOR OPTIMAL SPLIT FACTORS


CYCLE NO. =       0
     FUNCTION VALUE = -0.20861E-01
     SPLIT FACTORS  =    0.50000
                         0.80000
                         0.90000
                         0.65000
                         0.05000

CYCLE NO. =       1
     FUNCTION VALUE = -0.23863E-01
     SPLIT FACTORS  =    0.62800
                         0.92800
                         1.00000
                         0.52200
                         0.0

CYCLE NO. =       2
     FUNCTION VALUE = -0.25713E-01
     SPLIT FACTORS  =    0.82000
                         1.00000
                         1.00000
                         0.33000
                         0.0

CYCLE NO. =       3
     FUNCTION VALUE = -0.25724E-01
     SPLIT FACTORS  =    0.82000
                         1.00000
                         1.00000
                         0.07400
                         0.0

CYCLE NO. =       4
     FUNCTION VALUE = -0.26586E-01
     SPLIT FACTORS  =    0.75599
                         1.00000
                         1.00000
                         0.01000
                         0.0

CYCLE NO. =       5
     FUNCTION VALUE = -0.26591E-01
     SPLIT FACTORS  =    0.62799
                         1.00000
                         1.00000
                         0.01000
                         0.0

CYCLE NO. =       6
CONTINUATION NO IMPROVEMENT, RETREAT, PATTERN DESTROYED
     FUNCTION VALUE = -0.26750E-01
     SPLIT FACTORS  =    0.69199
                         1.00000
                         1.00000
                         0.07400
                         0.0
```

TABLE F-2 Continued

```
CYCLE NO. =      7
HALVE STEP SIZE
     FUNCTION VALUE = -0.26750E-01
     SPLIT FACTORS  =    0.69199
                         1.00000
                         1.00000
                         0.07400
                         0.0

CYCLE NO. =      8
     FUNCTION VALUE = -0.26669E-01
     SPLIT FACTORS  =    0.75599
                         1.00000
                         1.00000
                         0.07400
                         0.0

CYCLE NO. =      9
CONTINUATION NO IMPROVEMENT, RETREAT, PATTERN DESTROYED
     FUNCTION VALUE = -0.26783E-01
     SPLIT FACTORS  =    0.72399
                         1.00000
                         1.00000
                         0.07400
                         0.0

CYCLE NO. =     10
HALVE STEP SIZE
     FUNCTION VALUE = -0.26783E-01
     SPLIT FACTORS  =    0.72399
                         1.00000
                         1.00000
                         0.07400
                         0.0

CYCLE NO. =     11
     FUNCTION VALUE = -0.26685E-01
     SPLIT FACTORS  =    0.75599
                         1.00000
                         1.00000
                         0.10600
                         0.0

CYCLE NO. =     12
CONTINUATION NO IMPROVEMENT, RETREAT, PATTERN DESTROYED
     FUNCTION VALUE = -0.26805E-01
     SPLIT FACTORS  =    0.73999
                         1.00000
                         1.00000
                         0.09000
                         0.0

CYCLE NO. =     13
HALVE STEP SIZE
     FUNCTION VALUE = -0.26805E-01
     SPLIT FACTORS  =    0.73999
                         1.00000
                         1.00000
                         0.09000
                         0.0
```

TABLE F-2 Continued

```
CYCLE NO. =    14
     FUNCTION VALUE = -0.26679E-01
     SPLIT FACTORS  =    0.75599
                         1.00000
                         1.00000
                         0.09000
                         0.0

CYCLE NO. =    15
CONTINUATION NO IMPROVEMENT, RETREAT, PATTERN DESTROYED
     FUNCTION VALUE = -0.26807E-01
     SPLIT FACTORS  =    0.74799
                         1.00000
                         1.00000
                         0.09000
                         0.0

CYCLE NO. =    16
HALVE STEP SIZE
     FUNCTION VALUE = -0.26807E-01
     SPLIT FACTORS  =    0.74799
                         1.00000
                         1.00000
                         0.09000
                         0.0

CYCLE NO. =    17
     FUNCTION VALUE = -0.26805E-01
     SPLIT FACTORS  =    0.73999
                         1.00000
                         1.00000
                         0.09000
                         0.0

CYCLE NO. =    18
CONTINUATION NO IMPROVEMENT, RETREAT, PATTERN DESTROYED
     FUNCTION VALUE = -0.26811E-01
     SPLIT FACTORS  =    0.74399
                         1.00000
                         1.00000
                         0.09000
                         0.0

CYCLE NO. =    19
HALVE STEP SIZE
     FUNCTION VALUE = -0.26811E-01
     SPLIT FACTORS  =    0.74399
                         1.00000
                         1.00000
                         0.09000
                         0.0

CYCLE NO. =    20
     FUNCTION VALUE = -0.26807E-01
     SPLIT FACTORS  =    0.74799
                         1.00000
                         1.00000
                         0.09000
                         0.0
```

TABLE F-2 Continued

```
CYCLE NO. =    21
CONTINUATION NO IMPROVEMENT, RETREAT, PATTERN DESTROYED
     FUNCTION VALUE = -0.26814E-01
     SPLIT FACTORS  =    0.74599
                         1.00000
                         1.00000
                         0.09000
                         0.0

CYCLE NO. =    22
HALVE STEP SIZE
     FUNCTION VALUE = -0.26814E-01
     SPLIT FACTORS  =    0.74599
                         1.00000
                         1.00000
                         0.09000
                         0.0

CYCLE NO. =    23
     FUNCTION VALUE = -0.26807E-01
     SPLIT FACTORS  =    0.74799
                         1.00000
                         1.00000
                         0.09000
                         0.0

CYCLE NO. =    24
CONTINUATION NO IMPROVEMENT, RETREAT, PATTERN DESTROYED
     FUNCTION VALUE = -0.26815E-01
     SPLIT FACTORS  =    0.74699
                         1.00000
                         1.00000
                         0.09000
                         0.0

CYCLE NO. =    25
CONTINUATION NO IMPROVEMENT, RETREAT, PATTERN DESTROYED
     FUNCTION VALUE = -0.26815E-01
     SPLIT FACTORS  =    0.74699
                         1.00000
                         1.00000
                         0.09000
                         0.0

SEARCH COMPLETED
```

TABLE F-3

Verification, Pattern Search Solution

Variable Split Factors

No.	Recovery Factor
1	a(i,11,5)
2	a(i,12,3)
3	a(i,13,1)
4	a(i,14,1)
5	a(i,15,1)

Alternate Initial Conditions Tried

Case No.	Split Factor				
	1	2	3	4	5
1	0.0	0.0	0.0	0.0	0.0
2	1.0	1.0	1.0	1.0	1.0
3	1.0	0.0	0.0	1.0	0.0
4	1.0	1.0	1.0	0.0	0.0
5	0.0	1.0	1.0	1.0	0.0

Results - Identical for All Cases

objective function = -0.02681

a(i,11,5) = 0.747

a(i,12,3) = 1.0

a(i,13,1) = 1.0

a(i,14,1) = 0.088

a(i,15,1) = 0.0

3. DOCUMENTATION

3.1 Pattern Search - BPSH,BPSOUT

Subroutine BPSH performs a Hooke-Jeeves pattern search, as described by Wilde and Beightler (20), modified to allow for the specification of upper and lower bounds. Possible areas of difficulty are saddle points and resolution valleys.

BPSH seeks to minimize a nonlinear function of from one to five variables. It requires two additional subroutines: FCT, for the function evaluation given a specified argument, and BPSOUT for intermediate communication of search progress if desired. The variables in the parameter list are described below; those which must be defined prior to calling BPSH are marked with an asterisk.

FCT* - name of external subroutine used for function evaluation. It calculates the function value F,

given an N dimensional argument vector ARG.
Parameter list: (F,ARG,N).

B* - initial argument vector, dimension N
- on return, B(1) holds min. function value

DX* - initial step size vector, dimension N

T - temporary head vector
- on return, argument vector corresponding to min. function value, dimension N

BND* - lower and upper bounds on argument, dimension 2 x N

EPS* - test value representing expected absolute error

N* - number of variables

MIT* - maximum no. of iterations (cycles)

IER - convergence flag
= -1 , no convergence in MIT iterations
= 0 , search continuing
> 0 , convergence in IER iterations

Subroutine BPSOUT is used to print the status of the search as it progresses if desired. The parameter list requires information as follows:

F - current function value

B - current search base (initial temporary head)

IT - current iteration number
= 0 prior to first search

kk - flag indicating status of pattern

= 1 - continuing as usual

= 2 - interrupted, on first interruption pattern is destroyed, subsequent consecutive interruptions, stepsize is halved.

k - flag indicating step size has been halved

= 0 no change

= 1 halved

IER - as above

3.2 VAL,STPRNT

Subroutine VAL calculates an optimal objective function value, F, for the linear optimization problem resulting from specifying the NN variable split factors s(I). In addition to the NN, and S(I), available from the parameter list, the variables in COMMON block /V/ must be specified as follows:

IV - for I = 1,2, ...NN

IV(I,1) = unit no. of Ith stream splitter

IV(I,2) = unit no. destination of first stream

IV(I,3) = unit no. destination of second stream

NFIN - print code

NFIN = -1 print L:P. error messages only

NFIN = 0 no output

NFIN = 1	print solutions, all system variables
NFIN > 1	print solution, decision variables only.

Subroutine STPRNT calculates and prints the dependent system variable values if required.

Other subroutines required are documented in Appendices D and E.

3.3 Mainline -- Deterministic Problem

This program solves the deterministic problem using a combination of pattern search and linear programming as presented in Chapter III, section C. The input data required are: data for pattern search initialization, listed in table F-1, defined in sections 3.1, 3.2; variable names for LPSOL, 21 are required similar to those listed in table D-1; and model data listed in table C-1, defined in table C-3.

The mainline prints a summary of initial conditions and BPSOUT summarizes intermediate search results - these are listed in table F-2. The optimal solution summary appears as table 9.

Subroutines required are documented as follows: INPUT - Appendix C; INIT - Appendix E; VAL,BPSH - Appendix F.

4. VERIFICATION OF SOLUTION

The solution was verified by solving the deterministic problem starting from a variety of initial conditions (split factors). The results are summarized in table F-3. As can be seen, a(i,14,1) is consistently 0.088 rather than 0.09 as above. This is probably the result of a resolution valley - a supposition supported by the low sensitivity shown by this factor in the sensitivity analysis.

APPENDIX G

SENSITIVITY ANALYSIS

Table of Contents

LIST OF TABLES

```
      C     MAINLINE -- S.A.

C     MAINLINE -- S.A.   SENSITIVITY ANALYSIS

C     PURPOSE
C         TO PERFORM A STANDARD SENSITIVITY ANALYSIS BY
C         EVALUATING THE EFFECT OF SPECIFIED PARAMETER OR
C         SYSTEM VARIABLE PERTURBATIONS ON OBJECTIVE
C         FUNCTION VALUE
C
C     REMARKS
C         DETAILED PERTURBATION RESULTS ARE WRITTEN ON
C         LOGICAL UNIT 1.
C
C     SUBROUTINES REQUIRED
C         INPUT - IMINP1, MINP1, MINP2, MINP3
C         INIT, COEFF, CONT2
C         CANON, TPHSIM, LPSOL
C         OUTPUT
C         SIN, (SOUT)
C
C  ************************************************************
C
C  SPECIFICATION - TRANSFORMATION EQUATION COEFFICIENTS

      REAL BIY(4,15,11),DBIY(4,15,11)

C  SPECIFICATION - MODEL DATA

      COMMON A(4,15,15),Y(4,15,11),D(4,6,15),CPROD(10,4,6),
     1CINS(10,4,15),CREDVA(10,11),FDCOST(4,11),
     1OPCOST(4,15),VALPRO(4,6),CON(4),RHS(10),RNGE(10),
     1GBND(4,15,2),PBND(4,6,2),FBND(11,2),SEL,NCODE(10),
     1NCOMP,NPRO,NG,NF,NCON,NB,NR

C  SPECIFICATION - L.P. DATA

      COMMON/LP/ C(30,60),RH(30),CZ(25),VNAME(25,3),
     1X(25),NC(30),M,N,NBAS(60),NPH,NS


    1 FORMAT(3A4)
    2 FORMAT('1',/////12X,'REFERENCE PROBLEM FROM EXPECTED
     * DATA')
    3 FORMAT('1',/////,12X,'PERTURBED PROBLEM - PERTURBATION
     * =',I3)
    4 FORMAT(/////,12X,'THE OPTIMAL SOLUTION TO THE
     * PERTURBED',
     1' PROBLEM IS')
    6 FORMAT(///12X,'THE OPTIMAL SOLUTION IS')

C  INITIALIZE COUNTERS
```

```
      C     MAINLINE -- S.A.  ...(CONT'D)

      NPERT=0
      NL=0

C  INPUT, INITIALIZATION OF MODEL DATA

      CALL INPUT
      CALL INIT

C  INPUT VARIABLE NAMES FOR LPSOL

      DO 5 J=1,NF
      READ(5,1)(VNAME(J,K),K=1,3)
    5 CONTINUE

C  SET UP AND SOLVE REFERENCE PROBLEM

      WRITE(6,2)
      CALL COEFF(BIY,DBIY)
      CALL CONT2(1,BIY,DBIY)
      CALL CANON
      CALL TPHSIM
      WRITE(6,6)
      CALL LPSOL(1)
      RVAL=RH(M+1)

C  WRITE SUMMARY TITLE BLOCK

      CALL OUTPUT(RVAL,OVAL,NL,NPERT)

C  BEGIN PROBLEM PERTURBATION EVALUATION

   15 NPERT=NPERT+1
      WRITE(6,3) NPERT

C  READ PERTURBATION SPECIFICATION, MODIFY MODEL DATA
C  ACCORDINGLY

      CALL SIN(&1000)

C  SET UP AND SOLVE PERTURBED PROBLEM

      CALL COEFF(BIY,DBIY)
      CALL CONT2(1,BIY,DBIY)
      CALL CANON
      CALL TPHSIM
      WRITE(6,4)
      CALL LPSOL(1)
      OVAL=RH(M+1)

C  ENTER RESULTS IN SUMMARY TABLE
```

```
          C     MAINLINE -- S.A.  ...(CONT'D)

      CALL OUTPUT(RVAL,OVAL,NL,NPERT)

C  RETURN MODEL DATA TO REFERENCE STATA

      CALL SOUT

C  NEXT PERTURBATION

      GOTO 15

C  ALL PERTURBATIONS ANALYSED, FINISH SUMMARY PRINT
C  (ALTERNATE RETURN FROM SIN)

 1000 NPERT=0
      CALL OUTPUT(RVAL,OVAL,NL,NPERT)
      STOP
      END
```

```
           C     SUBROUTINE SIN

C     SUBROUTINE SIN

C     PURPOSE
C         TO READ PERTURBATION SPECIFICATION, MODIFY MODEL
C         DATA ACCORDINGLY, AND MAKE APPROPRIATE RETURN WHEN
C         PERTURBATIONS ARE FINISHED.
C
C         TO RETURN MODEL DATA TO REFERENCE STATE, AFTER
C         EACH PERTURBATION IS ANALYSED, VIA SECOND
C         ENTRY POINT SOUT
C
C     REMARKS
C         PERTURBATIONS ARE SUMMARIZED ON LOGICAL UNIT 6
C
C   ***********************************************************

      SUBROUTINE SIN(*)


      DIMENSION NT(10),V(10),L(3,10)
      COMMON A(4,15,15),Y(4,15,11),D(4,6,15),CPROD(10,4,6),
     1CINS(10,4,15),CREDVA(10,11),FDCOST(4,11),
     1OPCOST(4,15),VALPRO(4,6),CON(4),RHS(10),RNGE(10),
     1GBND(4,15,2),PBND(4,6,2),FBND(11,2),SEL,NCODE(10),
     1NCOMP,NPRO,NG,NF,NCON,NB,NR


    1 FORMAT(I10,F10.4,3I10)
    2 FORMAT(//12X,'END OF DATA')
    3 FORMAT(////T24,'PROCESS DATA CHANGES',//T16,'CHANGE',
     1T35,'FROM',T47,'TO'/)
    4 FORMAT(12X,'A(',I2,',',I2,',',I2,')',1X,2F15.4)
    5 FORMAT(12X,'D(',I2,',',I2,',',I2,')',1X,2F15.4)
    6 FORMAT(15X,'CON(',I1,')',3X,2F15.4)

C  READ PERTURBATION SPECIFICATION - BASE PARAMETER

      READ(5,1) N,VAL,I,J,K
      IF(N.NE.0) WRITE(6,3)
      M=0

C  MAKE APPROPRIATE MODIFICATIONS TO MODEL DATA

      GOTO(10,20,30),N
      WRITE(6,2)

C  NO MORE PERTURBATION DATA, MAKE APPROPRIATE RETURN

      RETURN 1
```

```
      C     SUBROUTINE SIN     ...(CONT'D)
C  MODIFY A RECOVERY FACTOR

   10 V(M+1)=A(I,J,K)
      A(I,J,K)=VAL
      WRITE(6,4) I,J,K,V(M+1),VAL
      GOTO 40

C  MODIFY A PRODUCT RECOVERY FACTOR

   20 V(M+1)=D(I,J,K)
      D(I,J,K)=VAL
      WRITE(6,5) I,J,K,V(M+1),VAL
      GOTO 40

C  MODIFY A REACTION CONVERSION FACTOR

   30 V(M+1)=CON(I)
      CON(I)=VAL
      WRITE(6,6) I,V(M+1),VAL
   40 M=M+1
      L(1,M)=I
      L(2,M)=J
      L(3,M)=K
      NT(M)=N

C  READ PERTURBATION SPECIFICATION  - COUPLED PARAMETER(S)

      READ(5,1) N,VAL,I,J,K

C  MAKE APPROPRIATE MODIFICATIONS TO MODEL DATA

      GOTO(10,20,30),N

C  PERTURBATION COMPLETE

      RETURN

C  MODEL DATA RESTORATION - RETURN MODEL DATA TO
C  REFERENCE STATE

      ENTRY SOUT
      DO 80 JJ=1,M
      N=NT(JJ)
      I=L(1,JJ)
      J=L(2,JJ)
      K=L(3,JJ)
      GOTO(50,60,70),N
   50 A(I,J,K)=V(JJ)
      GOTO 80
   60 D(I,J,K)=V(JJ)
```

C SUBROUTINE SIN ...(CONT'D)

```
      GOTO 80
   70 CON(I)=V(JJ)
   80 CONTINUE
      RETURN
      END
```

C SUBROUTINE OUTPUT

```
C     SUBROUTINE OUTPUT
C
C     PURPOSE
C          TO SUMMARIZE THE RESULTS OF THE SENSITIVITY
C          ANALYSIS IN TABLE FORM
C
C     REMARKS
C          THE SUMMARY IS WRITTEN ON LOGICAL UNIT 1
C
C   ****************************************************************

      SUBROUTINE OUTPUT(DVAL,OVAL,NL,NPERT)


    1 FORMAT(15A4)
    2 FORMAT('1'/////,13X,15A4)
    3 FORMAT(/16X,'THE OPTIMAL SOLUTION TO THE '
     2,'DETERMINISTIC PROBLEM IS USED'/16X,'AS THE REFERENCE
     * ',
     3'STRATEGY.'//20X,'DVAL = REFERENCE OBJECTIVE ',
     4'FUNCTION VALUE'/25X,'= ',F12.6//20X,'OVAL = OBJECTIVE
     * ',
     5'FUNCTION VALUE FOR THE PERTURBEB'/27X,'PROBLEM ',
     6'UNDER THE OPTIMAL STRATEGY FOR'/27X,'THAT PROBLEM',
     7//20X,'NPERT = PERTURBATION NUMBER')
    4 FORMAT (///22X,'NPERT',7X,'OVAL',9X,'OVAL-DVAL',5X,'%
     * CHANGE'/)
    5 FORMAT('1',//)
    6 FORMAT(19X,I6,2X,3F13.6)
    7 FORMAT('1FINI')


      REAL DVAL,OVAL,TITLE(15),VAL(2)
      INTEGER NL,NPERT

C   WRITE SUMMARY TITLE ON FIRST ENTRY

      IF(NL.NE.0) GOTO 100
      READ(5,1) TITLE
      WRITE(1,2) TITLE
      WRITE(1,3) DVAL
      WRITE(1,4)

C   INITIALIZE LINE COUNT

      NL=26
      GOTO 150

C   WRITE PERTURBATION RESULTS IF NOT LAST ENTRY
```

```
          C    SUBROUTINE OUTPUT  ...(CONT'D)

  100 IF(NPERT.EQ.0) GOTO 170

C  IF NECESSARY, SPACE TO NEW PAGE

      IF(NL-59) 120,120,110
  110 WRITE(1,5)
      WRITE(1,4)
      NL=11

C  INTERPRET, WRITE RESULTS

  120 VAL(1) = OVAL-DVAL
      VAL(2) = VAL(1)*100./DVAL
      WRITE(1,6) NPERT,OVAL,VAL
      NL=NL+1
  150 CONTINUE

C  RESULT PRINT FINISHED

      RETURN

C  LAST ENTRY, SPACE TO NEW PAGE

  170 WRITE(1,7)
      RETURN
      END
```

```
          C     MAINLINE -- R.A.

C     MAINLINE -- R.A.   RANGE ANALYSIS

C     PURPOSE
C         TO IDENTIFY THOSE CRITICAL SPLIT FACTORS WHICH
C         SHOW AN APPRECIABLE RANGE OF VARIATION IN THEIR
C         OPTIMAL VALUES WHEN CRITICAL SYSTEM PARAMETERS
C         ARE PERTURBED
C
C     REMARKS
C         DETAILED RESULTS ARE WRITTEN ON LOGICAL UNIT 6
C         A SUMMARY IS WRITTEN ON LOGICAL UNIT 1.
C
C     SUBROUTINES REQUIRED
C         INPUT - IMINP1, MINP1, MINP2, MINP3
C         INIT
C         VAL - COEFF, CONT2, CANON,  PHSIM, LPSOL, STPRNT
C         SOP
C         SIN, (SOUT)
C         BPSH - VAL (DUMMY NAME FCT)8 BPSOUT
C
C  ************************************************************
C
C  SPECIFICATION OF VARIABLES REQUIRED FOR BPSH

      EXTERNAL VAL
      REAL B(5),T(5),DX(5),DXM(5),BND(2,5)

C  COMMON SPECIFICATION, VARIABLES REQUIRED FOR VAL

      COMMON /V/ IV(5,3),NFIN
      COMMON /LP/ C(30,60),RH(30),CZ(25),VNAME(25,3),
     UX(25),NC(30),M,NCOL,NBAS(60),NPH,NS


    1 FORMAT(3A4)
    2 FORMAT('1',////12X,'REFERENCE PROBLEM FROM EXPECTED
     * DATA')
    3 FORMAT('1',////,12X,'PERTURBED PROBLEM - PERTURBATION
     * =',I3)
    4 FORMAT(///,12X,'AFTER',I4,' CYCLES'//12X,'THE OPTIMAL
     * SOLUTION TO'
     1,' THE PERTURBED PROBLEM IS')
    6 FORMAT(///12X,'THE OPTIMAL SOLUTION IS')
   11 FORMAT(3I10)
   12 FORMAT(5G15.5)
   13 FORMAT(//12X,'TOO MANY ITERATIONS')
   14 FORMAT(13X,'SPLIT FACTOR',I2,' = ',F8.5)
   16 FORMAT(//)

C  INITIALIZE COUNTERS
```

```
      C     MAINLINE -- R.A.  ...(CONT'D)
      NPERT=0
      NL=0
C   INPUT, INITIALIZATION OF MODEL DATA
      CALL INPUT
      CALL INIT
C   INPUT OF VARIABLE NAMES FOR LPSOL
      DO 5 J=1,11
      READ(5,1)(VNAME(J,K),K=1,3)
    5 CONTINUE
C   INPUT OF DATA FOR PATTERN SEARCH INITIALIZATION
      READ(5,11) N,MIT
      DO 7 J=1,N
    7 READ(5,11) (IV(J,K),K=1,3)
      READ(5,12) (B(J),J=1,N)
      READ(5,12) (DX(J),J=1,N)
      READ(5,12) (DXM(J),J=1,N)
      DO 8 K=1,2
    8 READ(5,12) (BND(K,J),J=1,N)
      READ(5,12) EPS
C   EVALUATE REFERENCE PROBLEM, PRINT RESULTS
      WRITE(6,2)
      WRITE(6,6)
      NFIN=1
      CALL VAL(RVAL,B,N)
      WRITE(6,16)
      WRITE(6,14) (J,B(J),J=1,N)
C   WRITE SUMMARY TITLE
      CALL SOP(T,RVAL,OVAL,NL,NPERT,N)
C   BEGIN PROBLEM PERTURBATION, EVALUATION
   15 NPERT=NPERT+1
      WRITE(6,3) NPERT
C   READ PERTURBATION SPECIFICATION, MODIFY MODEL DATA
C   ACCORDINGLY
      CALL SIN(&1000)
C   SET UP AND SOLVE PERTURBED PROBLEM - FIND OPTIMAL SPLIT
```

```
          C     MAINLINE -- R.A.   ...(CONT'D)

C   FACTOR VALUES

        NFIN=0
        CALL BPSH(VAL,B,DX,DXM,T,BND,EPS,N,MIT,IER)
        IF(IER) 20,25,25
   20   WRITE(6,13)
        GOTO 30

C   PRINT DETAILED RESULTS

   25   WRITE(6,4) IER
   30   NFIN=2
        CALL VAL(OVAL,T,N)

C   ENTER RESULTS IN SUMMARY TABLE

        CALL SOP(T,RVAL,OVAL,NL,NPERT,N)

C   RESTORE MODEL DATA TO REFERENCE STATE

        CALL SOUT

C   EVALUATE NEXT PERTURBATION

        GOTO 15

C   ALL PERTURBATIONS ANALYSED, FINISH SUMMARY PRINT
C   (ALTERNATE RETURN FROM SIN)

 1000   NPERT=0
        CALL SOP(T,RVAL,OVAL,NL,NPERT,N)
        STOP
        END
```

```
        C     SUBROUTINE SOP

C     SUBROUTINE SOP

C     PURPOSE
C         TO SUMMARIZE THE RESULTS OF THE RANGE ANALYSIS IN
C         TABLE FORM
C
C     REMARKS
C         THE SUMMARY IS WRITTEN ON LOGICAL UNIT 1
C
C   ***********************************************************

      SUBROUTINE SOP(T,DVAL,OVAL,NL,NPERT,N)


    1 FORMAT(15A4)
    2 FORMAT('1'/////,13X,15A4)
    3 FORMAT(/16X,'THE OPTIMAL SOLUTION TO THE '
     2,'DETERMINISTIC PROBLEM IS USED'/16X,'AS THE REFERENCE
     * ',
     3'STRATEGY.'//20X,'DVAL = REFERENCE OBJECTIVE ',
     4'FUNCTION VALUE'/25X,'= ',F12.6//20X,'OVAL = OBJECTIVE
     * ',
     5'FUNCTION VALUE FOR THE PERTURBEB'/27X,'PROBLEM ',
     6'UNDER THE OPTIMAL STRATEGY FOR'/27X,'THAT PROBLEM',
     7//20X,'NPERT = PERTURBATION NUMBER')
    4 FORMAT(///22X,'NPERT',3X,'% CHANGE',2X,2('S.F.',I1
     *,3X))
    5 FORMAT('1',//)
    6 FORMAT(19X,I6,F12.2,5F8.3)
    7 FORMAT('1FINI')
    8 FORMAT(/17X,'% CHANGE = % CHANGE IN OBJECTIVE FUNCTION
     * VALUE')
    9 FORMAT(///22X,'SPLIT FACTOR',3X,'MEAN',4X,'VARIANCE'
     */(22X,I8,
     1F13.3,E12.2))


      REAL DVAL,OVAL,TITLE(15),T(1),SM(10),SV(10)
      INTEGER NL,NPERT

C   WRITE SUMMARY TITLE BLOCK ON FIRST ENTRY, INITIALIZE
C   WORK SPACE

      IF(NL.NE.0) GOTO 100
      READ(5,1) TITLE
      WRITE(1,2) TITLE
      WRITE(1,3) DVAL
      WRITE(1,8)
      WRITE(1,4)(I,I=1,N)
      DO 90 J=1,N
```

```
      C     SUBROUTINE SOP    ...(CONT'D)

      SM(J)=0.0
   90 SV(J)=0.0
      NL=1
      GOTO 150

C  INTERPRET AND WRITE INTERMEDIATE RESULTS IF NOT LAST
C  ENTRY

  100 IF(NPERT.EQ.0) GOTO 170
  120 VAL=(OVAL-DVAL)*100./DVAL
      WRITE(1,6) NPERT,VAL,(T(J),J=1,N)

C  PERFORM SUMMATIONS FOR STATISTICS

      DO 130 J=1,N
      SM(J)=SM(J)+T(J)
  130 SV(J)=SV(J)+T(J)*T(J)
      NL=NPERT
  150 CONTINUE
      RETURN

C  LAST ENTRY, CALCULATE AND WRITE STATISTICS ON SPLIT
C  FACTOR VALUE, SPACE TO A NEW PAGE

  170 DO 180 J=1,N
      SM(J)=SM(J)/NL
  180 SV(J)=(SV(J)-NL*SM(J)*SM(J))/(NL-1)
      WRITE(1,9)(J,SM(J),SV(J),J=1,N)
      WRITE(1,5)
      RETURN
      END
```

C SUBROUTINE BPSOUT

```
C     SUBROUTINE BPSOUT

C     DUMMY OUTPUT ROUTINE REQUIRED BY BPSH
C

      SUBROUTINE BPSOUT(F,B,N,IT,KK,K,IER)
      REAL B(1)
      RETURN
      END
```

2. TABLES

TABLE G - 1

SENSITIVITY ANALYSIS DATA

PARAMETER PERTURBATIONS

BUTADIENE AREA - PARAMETER SENSITIVITY

1	0.97	1	2	3
2	0.03	1	1	2
0	.			
1	0.87	1	2	3
2	0.13	1	1	2
0	.			
1	0.20	2	1	2
1	0.80	2	1	8
00	.			
1	0.10	2	1	2
1	0.90	2	1	8
0	.			
1	0.95	2	2	3
2	0.05	2	1	2
0	.			
1	0.85	2	2	3
2	0.15	2	1	2
0	.			
1	0.20	3	2	3
2	0.80	3	1	2
0	.			
1	0.10	3	2	3
2	0.90	3	1	2
0	.			
1	0.95	3	3	4
0	.			
1	0.85	3	3	4
0	.			
1	0.95	4	3	4
0	.			
1	0.85	4	3	4
0	.			
1	0.975	1	4	11
1	0.975	2	4	11
1	0.975	3	4	11
1	0.975	4	4	11
2	0.025	1	2	4
2	0.025	2	2	4
2	0.025	3	2	4
2	0.025	4	2	4
0	.			
1	0.925	1	4	11
1	0.925	2	4	11
1	0.925	3	4	11

TABLE G - 1 ...CONT'D

1	0.925	4	4	11
2	0.075	1	2	4
2	0.075	2	2	4
2	0.075	3	2	4
2	0.075	4	2	4
0				
1	0.075	4	5	12
2	0.925	4	3	5
0	.			
1	0.025	4	5	12
2	0.975	4	3	5
0	.			
1	0.25	1	6	7
1	0.75	1	6	14
0	.			
1	0.15	1	6	7
1	0.85	1	6	14
0	.			
1	0.25	2	6	7
1	0.75	2	6	14
0	.			
1	0.15	2	6	7
1	0.85	2	6	14
0	.			
1	0.975	4	6	7
1	0.025	4	6	14
0	.			
1	0.925	4	6	7
1	0.075	4	6	14
0	.			
1	0.075	4	7	15
2	0.925	4	4	7
0	.			
1	0.025	4	7	15
2	0.975	4	4	7
0	.			
1	0.85	2	8	9
2	0.05	2	5	8
0	.			
1	0.75	2	8	9
2	0.15	2	5	8
0	.			
1	0.15	2	9	1
1	0.85	2	9	10
0	.			
1	0.05	2	9	1
1	0.95	2	9	10
0	.			
1	0.15	2	10	1
2	0.80	2	6	10

TABLE G - 1 ...CONT'D

0	.			
1	0.05	2	10	1
2	0.90	2	6	10
0	.			
3	-0.475	1		
3	0.475	4		
0				
3	-0.325	1		
3	0.325	4		
0				
0				

TABLE G - 2

SENSITIVITY ANALYSIS DATA

OPTIMAL SPLIT FACTOR PERTURBATIONS

BUTADIENE AREA - SPLIT FACTOR SENSITIVITY

1	0.80	1	11	5
1	0.80	2	11	5
1	0.80	3	11	5
1	0.80	4	11	5
1	0.20	1	11	6
1	0.20	2	11	6
1	0.20	3	11	6
1	0.20	4	11	6
0				
1	0.7	1	11	5
1	0.7	2	11	5
1	0.7	3	11	5
1	0.7	4	11	5
1	0.3	1	11	6
1	0.3	2	11	6
1	0.3	3	11	6
1	0.3	4	11	6
0				
1	0.85	1	11	5
1	0.85	2	11	5
1	0.85	3	11	5
1	0.85	4	11	5
1	0.15	1	11	6
1	0.15	2	11	6
1	0.15	3	11	6
1	0.15	4	11	6
0				
1	0.65	1	11	5
1	0.65	2	11	5
1	0.65	3	11	5
1	0.65	4	11	5
1	0.35	1	11	6
1	0.35	2	11	6
1	0.35	3	11	6
1	0.35	4	11	6
0				
1	0.95	1	12	3
1	0.95	2	12	3
1	0.95	3	12	3
1	0.95	4	12	3
1	0.05	1	12	13
1	0.05	2	12	13
1	0.05	3	12	13
1	0.0 05	4	12	13

TABLE G - 2 ...CONT'D

0				
1	0.9	1	12	3
1	0.9	2	12	3
1	0.9	3	12	3
1	0.9	4	12	3
1	0.1	1	12	13
1	0.1	2	12	13
1	0.1	3	12	13
1	0.1	4	12	13
0				
1	0.04	1	14	1
1	0.04	2	14	1
1	0.04	3	14	1
1	0.04	4	14	1
1	0.96	1	14	3
1	0.96	2	14	3
1	0.96	3	14	3
1	0.96	4	14	3
0				
1	0.14	1	14	1
1	0.14	2	14	1
1	0.14	3	14	1
1	0.14	4	14	1
1	0.86	1	14	3
1	0.86	2	14	3
1	0.86	3	14	3
1	0.86	4	14	3
0				
1	0.0	1	14	1
1	0.0	2	14	1
1	0.0	3	14	1
1	0.0	4	14	1
1	1.0	1	14	3
1	1.0	2	14	3
1	1.0	3	14	3
1	1.0	4	14	3
0				
1	0.19	1	14	1
1	0.19	2	14	1
1	0.19	3	14	1
1	0.19	4	14	1
1	0.81	1	14	3
1	0.81	2	14	3
1	0.81	3	14	3
1	0.81	4	14	3
0				
1	0.95	1	15	3
1	0.95	2	15	3
1	0.95	3	15	3
1	0.95	4	15	3

TABLE G - 2 ...CONT'D

1		0.05	1	15	1
1		0.05	2	15	1
1		0.05	3	15	1
1		0.05	4	15	1
0					
1	0.9		1	15	3
1	0.9		2	15	3
1	0.9		3	15	3
1	0.9		4	15	3
1		0.1	1	15	1
1		0.1	2	15	1
1		0.1	3	15	1
1		0.1	4	15	1
0					
0					

TABLE G - 3

RANGE ANALYSIS DATA

DATA FOR BPSH - PATTERN SEARCH

2	20	
11	5	6
12	3	13
0.74	1.0	
.02	.02	
.02	.02	
1.0	1.0	
0.0	0.0	
1.0 E-06		

PARAMETER PERTURBATIONS

BUTADIENE AREA - S.F. RANGE ANALYSIS

1	0.20	2	1	2
1	0.80	2	1	8
00	.			
1	0.10	2	1	2
1	0.90	2	1	8
0	.			
1	0.95	4	3	4
0	.			
1	0.85	4	3	4
0	.			
1	0.975	1	4	11
1	0.975	2	4	11
1	0.975	3	4	11
1	0.975	4	4	11
2	0.025	1	2	4
2	0.025	2	2	4
2	0.025	3	2	4
2	0.025	4	2	4
0	.			
1	0.925	1	4	11
1	0.925	2	4	11
1	0.925	3	4	11
1	0.925	4	4	11
2	0.075	1	2	4
2	0.075	2	2	4
2	0.075	3	2	4
2	0.075	4	2	4
0				
1	0.85	2	8	9
2	0.05	2	5	8
0	.			

TABLE G - 3 ...CONT'D

1	0.75	2	8	9
2	0.15	2	5	8
0	.			
1	0.15	2	9	1
1	0.85	2	9	10
0	.			
1	0.05	2	9	1
1	0.95	2	9	10
0	.			
1	0.15	2	10	1
2	0.80	2	6	10
0	.			
1	0.05	2	10	1
2	0.90	2	6	10
0	.			
0				

3. DOCUMENTATION

3.1 SIN

This subroutine reads data specifying a problem perturbation and modifies the model data (stored in unlabelled COMMON) accordingly. The original model data may be restored by entering the subroutine at its second entry point, SOUT (i.e. CALL SOUT)

On each perturbation data card, the following variables may be specified.

N - type of perturbation

= 0 end of data sub set 2nd consecutive
0 indicates end of perturbations

= 1 change A(I,J,k)

= 2 change D(I,J,k)

= 3 change CON(I)

VAL - new value for perturbed parameter

I,J,K - appropriate indices

When the end of the perturbation data set is indicated, an appropriate return is made to the calling program statement number specified in the parameter list.

3.2 OUTPUT, SOP

These subroutines summarize the results of the sensitivity analysis and range analysis. A title card

listing the heading to be printed on the summary is required from input data. The parameter list variables are:

DVAL - reference objective function value

OVAL - optimal objective function value, perturbed problem

NL - flag

= 0 for first perturbation

NPERT - No. of perturbation

= 0 for end of perturbations

<u>3.3 Mainline -- S.A.</u>

This program performs the sensitivity analyses described in chapter III, section D. The input data required are: model data similar to that listed in table C-1, defined in table C-3, but with optimal split factors; variable names for LPSOL, listed in table D-1; and sensitivity analysis data consisting of a title card and perturbation data. Sensitivity analysis data are listed in tables G-1 and G-2.

The sensitivity analysis summaries printed by OUTPUT appear as tables 11 and 13.

Subroutines required are documented as follows:

INPUT - Appendix C

INIT, COEFF, CONT2 - Appendix E

CANON, TPHSIM, LPSOL - Appendix D

SIN, OUTPUT, - Appendix G

3.4 Mainline -- R.A.

This program performs the range analysis described in chapter III, section D. The input data required are: model data similar to that listed in table C-1, defined in table C-3, but with optimal split factors; variable names for LPSOL, listed in table D-1; and range analysis data consisting of pattern search initialization data, a title card, and perturbation data. Range analysis data are listed in table G-3.

The range analysis summary printed by SOP appears as table 15.

Subroutines required are documented as follows:

INPUT - Appendix C

INIT - Appendix E

VAL,BPSH - Appendix F

SIN,SOP - Appendix G

BPSOUT for BPSH - Appendix G

APPENDIX H

EXPECTED COST OF UNCERTAINTY ESTIMATION

Table of Contents

LIST OF TABLES

```
          C     MAINLINE -- E.C.

C     MAINLINE -- E.C.  EXPECTED COST OF UNCERTAINTY

C     PURPOSE
C         TO ESTIMATE THE EXPECTED COST OF UNCERTAINTY
C         ABOUT SENSITIVE MODEL PARAMETERS
C
C     SUBROUTINES REQUIRED
C         INPUT - IMINP1, MINP1, MINP2, MINP3
C         INIT, INIT2
C         ECOST - EVAL, AVAL, GAUSS, REFEED, COEFF, CONT2,
C                 CANON, TPHSIM, LPSOL
C
C     REMARKS
C         DETAILED RESULTS APPEAR ON LOGICAL UNIT 6
C         SUMMARIES ARE WRITTEN ON LOGICAL UNIT 8.
C
C   ***********************************************************

      COMMON /EC/ RM(10),S(10),AT(10),SUM(5),STAT(6),ALPHA
     1,DELTA,CONF,NTS(10,8),NSF,MAXSAM,MINSAM,NSAM,IX,NREJ
      COMMON /LP/ CON(30,60),RHS(30),CZ(25),VNAME(25,3),
     1X(25),NCODE(30),M,N,NBAS(60),NPH,NS


    1 FORMAT(3A4)
    2 FORMAT(8I10)
    3 FORMAT(8F10.4)
    4 FORMAT(5G15.7)
    5 FORMAT('1'/////12X,'EVALUATION OF EXPECTED COST OF ',
     1'UNCERTAINTY'/12X,'BY MONTE CARLO SIMULATION')
    6 FORMAT(//12X,'THE SENSITIVE FACTORS ARE -'/)
    7 FORMAT(12X,'A(',I2,',',I2,',',I2,')',7X,'MEAN =',
     'F10.4,7X,'VARIANCE =',F10.4)
    8 FORMAT(12X,'D(',I2,',',I2,',',I2,')',7X,'MEAN =',
     'F10.4,7X,'VARIANCE =',F10.4)
    9 FORMAT(12X,'CON(',I2,')',11X,'MEAN =',F10.4,7X,
     1'VARIANCE =',F10.4)
   10 FORMAT(17X,'WITH DEPENDENT A(',I2,',',I2,',',I2,')')
   11 FORMAT(17X,'WITH DEPENDENT D(',I2,',',I2,',',I2,')')
   12 FORMAT(17X,'WITH DEPENDENT CON(',I2,')')
   13 FORMAT(///12X,'MAXIMUM NO. OF SAMPLES THIS RUN =',I5/
     112X,'ALPHA =',F10.4,/12X,'DELTA =',G13.4/12X,
     2'CONFIDENCE LEVEL =',F10.4)
   14 FORMAT(//12X,'THIS RUN IS A CONTINUATION'/12X,
     1'NO. OF SAMPLES ALREADY TAKEN =',I10)

C   INPUT OF MODEL DATA, INITIALIZATION FOR LP

      CALL INPUT
      CALL INIT
```

```
      C     MAINLINE -- E.C.  ...(CONT'D)

      CALL INIT2
      DO 20 J=1,11
   20 READ(5,1) (VNAME(J,K),K=1,3)

C  INPUT OF DATA INDICATING SENSITIVE PARAMETERS

      WRITE(8,5)
      WRITE(8,6)
      READ(5,2) NSF
      DO 40 J=1,NSF
      READ(5,2)(NTS(J,K),K=1,8)
      READ(5,3) RM(J),S(J),AT(J)
      K=NTS(J,1)
      GOTO (31,32,33,31,32),K
      GOTO 39
   31 WRITE(8,7) (NTS(J,K),K=2,4),RM(J),S(J)
      GOTO 34
   32 WRITE(8,8)(NTS(J,K),K=2,4),RM(J),S(J)
      GOTO 34
   33 WRITE(8,9) NTS(J,2),RM(J),S(J)
   34 K=NTS(J,5)
      GOTO (36,37,38,36,37),K
      GOTO 39
   36 WRITE(8,10) (NTS(J,K),K=6,8)
      GOTO 39
   37 WRITE(8,11)(NTS(J,K),K=6,8)
      GOTO 39
   38 WRITE(8,12) NTS(J,6)
   39 S(J)=S(J)**0.5
   40 CONTINUE

C  INPUT DATA FOR SIMULATION

      READ(5,2) MAXSAM,MINSAM
      READ(5,4) ALPHA,DELTA,CONF
      WRITE(8,13) MAXSAM,ALPHA,DELTA,CONF
      READ(5,2) NSAM,IX
      IF(NSAM.EQ.0) GOTO 50
      READ(5,4) SUM
      READ(5,4) STAT
      WRITE(8,14) NSAM

C  ESTIMATE EXPECTED COST OF UNCERTAINTY

   50 CALL ECOST
      STOP
      END
```

```
          C     SUBROUTINE ECOST

C     SUBROUTINE ECOST

C     PURPOSE
C         TO ESTIMATE THE EXPECTED COST OF UNCERTAINTY BY
C         MONTE CARLO SIMULATION.
C
C     REMARKS
C         A SUMMARY OF RESULTS IS WRITTEN ON LOGICAL UNIT 8
C
C     SUBROUTINES REQUIRED
C         EVAL
C
C  ***********************************************************

      SUBROUTINE ECOST


      COMMON /EC/ RM(10),S(10),AT(10),SUM(5),STAT(6),ALPHA
     1,DELTA,CONF,NTS(10,8),NSF,MAXSAM,MINSAM,NSAM,IX,NREJ


    1 FORMAT('1'///16X,'NO. OF',5X,'REGRET',11X,'COST',11X
     *,'COST',T93,
     1'EXP. COST',4X,'VARIANCE',3X,'PRECISION'/15X,'SAMPLE'
     *,20X,
     2'CERTAINTY',3X,'REF. STRATEGY',T92,'UNCERTAINTY'/)
    2 FORMAT(1H+,T90,3E12.2)
    3 FORMAT(2I10)
    4 FORMAT(5G15.7)
  106 FORMAT('1'/////12X,'RESULTS OF MONTE CARLO SIMULATION'
     */)
  107 FORMAT(12X,'NO. OF SAMPLES TAKEN =',I5/12X,'NO. OF
     * SAMPLES '
     2,'REJECTED =',I5)
  108 FORMAT(/12X,'EXPECTED COST OF UNCERTAINTY =',G11.3,'
     * VAR. =',
     2G10.2//12X,'EXPECTED COST, REF. STRATEGY =',G11.3,'
     * VAR. =',G10.2/
     212X,'EXPECTED COST, CERTAINTY =',G11.3,' VAR. ='
     *,G10.2)
  109 FORMAT(/12X,'PRECISION, COST OF UNCERTAINTY ESTIMATE
     * =',G10.2/
     112X,'CONFIDENCE LEVEL REQUESTED =',F6.3)
  110 FORMAT(//12X,'SIMULATION COMPLETED'/'1')
  111 FORMAT(/12X,'SAMPLES REQUIRED TO COMPLETE SIMULATION
     * =',I10/
     112X,'TOTAL SAMPLES REQUIRED =',I10/'1')

C  WRITE SUMMARY TITLE
```

```
      C     SUBROUTINE ECOST  ...(CONT'D)

      WRITE(8,1)

C  INITIALIZE COUNTERS, WORKSPACE

      NREJ=0
      I=0
      IF(NSAM) 8,8,30
    8 DO 10 J=1,5
   10 SUM(J)=0.0

C  BEGIN SAMPLING

   20 I=I+1
      NSAM=NSAM+1

C  GENERATE SAMPLE - REFERENCE VALUE, OPTIMAL VALUE

      CALL EVAL(RVAL,OVAL,&20)

C  PERFORM SUMMATIONS

      SUM(1)=SUM(1)+RVAL
      SUM(2)=SUM(2)+RVAL*RVAL
      SUM(3)=SUM(3)+OVAL
      SUM(4)=SUM(4)+OVAL*OVAL
      SUM(5)=SUM(5)+(RVAL-OVAL)**2

C  IF INSUFFICIENT SAMPLES TAKEN FOR CALCULATION OF
C  STATISTICS, SAMPLE AGAIN

      IF(NSAM.LT.MINSAM) GOTO 20

C  CALCULATE SAMPLE MEANS - REFERENCE VALUE, OPTIMAL VALUE,
C  EXPECTED COST - AND SAMPLE VARIANCE,EXPECTED COST

      STAT(3)=SUM(1)/NSAM
      STAT(5)=SUM(3)/NSAM
      STAT(1)=STAT(3)-STAT(5)
      STAT(2)=(SUM(5)-NSAM*STAT(1)*STAT(1))/(NSAM-1)

C  ESTIMATE ACCURACY, EXPECTED COST ESTIMATE

   30 TN=ALPHA*((STAT(2)/NSAM)**0.5)
      WRITE(8,2) STAT(1),STAT(2),TN

C  IF ACCURACY IS SUFFICIENT, SIMULATION IS  COMPLETED

      IF(DELTA.GE.TN) GOTO 35

C  IF NOT, AND MORE SAMPLES MAY BE TAKEN, SAMPLE AGAIN
```

```
         C     SUBROUTINE ECOST  ...(CONT'D)

      IF(I.LT.MAXSAM) GOTO 20

C   COMPLETE SIMULATION
C   CALCULATE SAMPLE VARIANCE - OPTIMAL VALUE, REFERENCE
C   VALUE

   35 STAT(4)=(SUM(2)-NSAM*STAT(3)*STAT(3))/(NSAM-1)
      STAT(6)=(SUM(4)-NSAM*STAT(5)*STAT(5))/(NSAM-1)

C   PUNCH INFORMATION FOR LATER CONTINUATION

      WRITE(7,3) NSAM,IX
      WRITE(7,4) SUM
      WRITE(7,4) STAT

C   WRITE SIMULATION SUMMARY

      WRITE(8,106)
      WRITE(8,107) NSAM,NREJ
      WRITE(8,108) STAT
      WRITE(8,109) TN, CONF

C   SIMULATION COMPLETE

      IF(DELTA.LT.TN) GOTO 40

C   YES, FINISHED

      WRITE(8,110)
      RETURN

C   NO, REQUIRED ACCURACY NOT OBTAINED
C   ESTIMATE, WRITE ADDITIONAL SAMPLES REQUIRED

   40 NSREQ=STAT(2)*(ALPHA/DELTA)**2
      NSREM=NSREQ-NSAM
      WRITE(8,111) NSREM,NSREQ
      RETURN
      END
```

```
          C     SUBROUTINE EVAL

C     SUBROUTINE EVAL

C     PURPOSE
C         TO GENERATE RANDOM VALUES OF THE REFERENCE
C         STRATEGY, THE OPTIMAL STRATEGY, AND THE REGRET
C         FOR USE IN THE MONTE CARLO SIMULATION ESTIMATE
C         OF THE EXPECTED COST OF UNCERTAINTY
C
C     REMARKS
C         SIMULATION RESULTS ARE WRITTEN IN DETAIL ON
C         LOGICAL UNIT 6 AND SUMMARIZED ON LOGICAL UNIT 8
C
C     SUBROUTINES REQUIRED
C         AVAL, REFEED, COEFF, CONT2, CANON, TPHSIM, LPSOL
C
C  ***********************************************************

      SUBROUTINE EVAL (RVAL,OVAL,*)


      REAL BIY(4,15,11),DBIY(4,6,11),RFEED(11)
      COMMON /LP/ CON(30,60),RHS(30),CZ(25),VNAME(25,3),
     1X(25),NCODE(30),M,N,NBAS(60),NPH,NS
      COMMON /EC/ RM(10),S(10),AT(10),SUM(5),STAT(6),ALPHA
     1,DELTA,CONF,NTS(10,8),NSF,MAXSAM,MINSAM,NSAM,IX,NREJ


    1 FORMAT('1',/////12X,'SAMPLE NO.',I5)
    2 FORMAT(////12X,'VALUE OF THE REFERENCE STRATEGY ='
     *,G12.4)
    3 FORMAT(//12X,'OPTIMAL STRATEGY FOR THIS STATE OF
     * NATURE -')
    4 FORMAT(//12X,'THE REGRET =',G12.4)
    5 FORMAT(//12X,'RANDOM VARIABLE VALUE OUT OF BOUNDS ',
     1'- SAMPLE REJECTED')
    6 FORMAT(12X,I8,2X,3(E13.4,2X,))

C  WRITE TITLE - SIMULATION RESULT PRINT

      WRITE(6,1) NSAM

CL GENERATE RANDOM SAMPLE - SENSITIVE PARAMETERS

      CALL AVAL(&100)

C  GENERATE TRANSFORMATION EQUATION COEFFICIENTS

      CALL COEFF(BIY,DBIY)

C  EVALUATE REFERENCE STRATEGY, PRINT RESULTS
```

```
      C     SUBROUTINE EVAL    ...(CONT'D)

      CALL REFEED(BIY,DBIY,RFEED)
      CALL CONT2(1,BIY,DBIY)
      RVAL=0
      DO 10 J=1,N
   10 RVAL=RVAL+RFEED(J)*CZ(J)
      WRITE(6,2) RVAL
      WRITE(6,3)

C  EVALUATE OPTIMAL STRATEGY, PRINT RESULTS

      CALL CANON
      CALL TPHSIM
      CALL LPSOL(1)
      NO=NCODE(M+2)
      OVAL=X(NO+1)

C  EVALUATE REGRET, PRINT

      REGRET=RVAL-OVAL
      WRITE(6,4) REGRET

C  WRITE SUMMARY FOR THIS SIMULATION

      WRITE(8,6) NSAM,REGRET,OVAL,RVAL

C  FINISHED

      RETURN

C  THE SAMPLE WAS REJECTED, CHANGE COUNTERS AND RETURN
C  APPROPRIATELY

  100 NSAM=NSAM-1
      NREJ=NREJ+1
      WRITE(6,5)
      RETURN 1
      END
```

```
      C     SUBROUTINE AVAL

C     SUBROUTINE AVAL

C     PURPOSE
C        TO GENERATE RANDOM SETS OF VALUES FOR THE
C        SENSITIVE PARAMETERS AND MODIFY MODEL DATA
C        ACCORDINGLY
C
C     REMARKS
C        THE PARAMETERS ARE WRITTEN ON LOGICAL UNIT 6
C
C     SUBROUTINES REQUIRED
C        GAUSS
C
C  ***************************************************************

      SUBROUTINE AVAL(*)


      COMMON /EC/ RM(10),S(10),AT(10),SUM(5),STAT(6),ALPHA
     1,DELTA,CONF,NTS(10,8),NSF,MAXSAM,MINSAM,NSAM,IX,NREJ
      COMMON A(4,15,15),Y(4,15,11),D(4,6,15),CPROD(10,4,6),
     1CINS(10,4,15),CREDVA(10,11),FDCOST(4,11),
     1OPCOST(4,15),VALPRO(4,6),CON(4),RHS(10),RNGE(10),
     1GBND(4,15,2),PBND(4,6,2),FBND(11,2),SEL,NCODE(10),
     1NCOMP,NPRO,NG,NF,NCON,NB,NR


    1 FORMAT(//12X,'SENSITIVE FACTORS'/)
    2 FORMAT(12X,'A(',I2,',',I2,',',I2,') =',F10.4)
    3 FORMAT(12X,'D(',I2,',',I2,',',I2,') =',F10.4)
    4 FORMAT(12X,'CON(',I2,')',5X,'=',F10.4)

C  WRITE TITLE

      WRITE(6,1)

C  GENERATE NSF SETS OF VALUES

      DO 50 J=1,NSF
      L=0
      IXI=IX
      RMM=RM(J)
      SD=S(J)

C  GENERATE A NORMAL RANDOM VARIABLE FROM THE REQUIRED
C  DISTRIBUTION

      CALL GAUSS(IXI,SD,RMM,AD)
      IX=IXI
```

```
          C      SUBROUTINE AVAL    ...(CONT'D)

C   IF THE GENERATED VARIABLE EXCEEDS ITS BOUNDS, THE SAMPLE
C   IS REJECTED

      IF((AD.GT.AT(J)).OR.(AD.LT.0.0)) RETURN 1

C   MODIFY BASE PARAMETER ACCORDINGLY

      K=NTS(J,1)
      GOTO (10,20,30,12,22),K
   10 A(NTS(J,L+2),NTS(J,L+3),NTS(J,L+4))=AD
      WRITE(6,2)(NTS(J,L+K),K=2,4),AD
      GOTO 40
   12 DO 15 M=1,4
      A(M,NTS(J,L+3),NTS(J,L+4))=AD
   15 WRITE(6,2) M,NTS(J,L+3),NTS(J,L+4),AD
      GOTO 40
   20 D(NTS(J,L+2),NTS(J,L+3),NTS(J,L+4))=AD
      WRITE(6,3)(NTS(J,L+K),K=2,4),AD
      GOTO 40
   22 DO 25 M=1,4
      D(M,NTS(J,L+3),NTS(J,L+4))=AD
   25 WRITE(6,3) M,NTS(J,L+3),NTS(J,L+4),AD
      GOTO 40
   30 CON(NTS(J,L+2))=AD
      WRITE(6,4) AD
      AD=AD+AT(J)

C   MODIFY COUPLED PARAMETER IF ANY

   40 IF(L.EQ.4) GOTO 50
      AD=AT(J)-AD
   41 L=4
      K=NTS(J,5)
      GOTO (10,20,30,12,22),K
   50 CONTINUE
      RETURN
      END
```

```
          C     SUBROUTINE INIT2

C     SUBROUTINE INIT2

C     PURPOSE
C         TO INITIALIZE MODEL DATA FOR EVALUATION OF
C         REFERENCE STRATEGY BY ADDING NECESSARY CONSTRAINT
C         INFORMATION
C
C  ***************************************************************

      SUBROUTINE INIT2


      REAL BIY(4,15,11),DBIY(4,6,11),RF(1),G(6),NCC(15)
      COMMON A(4,15,15),Y(4,15,11),D(4,6,15),CPROD(10,4,6),
     1CINS(10,4,15),CREDVA(10,11),FDCOST(4,11),
     1OPCOST(4,15),VALPRO(4,6),CON(4),RHS(10),RNGE(10),
     1GBND(4,15,2),PBND(4,6,2),FBND(11,2),SEL,NCODE(10),
     1NCOMP,NPRO,NG,NF,NCON,NB,NR

C  ADD CONSTRAINT 1

      CREDVA(NCON+1,6)=1.0
      RHS(NCON+1)=FBND(6,1)

C  ADD CONSTRAINT 2

      CREDVA(NCON+2,3) =1.0
      CREDVA(NCON+2,4) =1.0
      CREDVA(NCON+2,5) =1.0
      CREDVA(NCON+2,7) =1.0
      CREDVA(NCON+2,8) =1.0
      CREDVA(NCON+2,10)=1.0
      CREDVA(NCON+2,11)=1.0
      RETURN
      END
```

C   SUBROUTINE REFEED

```
C     SUBROUTINE REFEED

C     PURPOSE
C          TO EVALUATE THE REFERENCE STRATEGY GIVEN
C          TRANSFORMATION EQUATION COEFFICIENTS
C
C     SUBROUTINES REQUIRED
C          CONT2, CANON, TPHSIM, LPSOL
C
C   ***********************************************************

      SUBROUTINE REFEED(BIY,DBIY,RF)


      COMMON A(4,15,15),Y(4,15,11),D(4,6,15),CPROD(10,4,6),
     1CINS(10,4,15),CREDVA(10,11),FDCOST(4,11),
     1OPCOST(4,15),VALPRO(4,6),CON(4),RHS(10),RNGE(10),
     1GBND(4,15,2),PBND(4,6,2),FBND(11,2),SEL,NCODE(10),
     1NCOMP,NPRO,NG,NF,NCON,NB,NR
      COMMON/LP/ C(30,60),RH(30),CZ(25),VNAME(25,3),
     1X(25),NC(30),M,N,NBAS(60),NPH,NS
      REAL BIY(4,15,11),DBIY(4,6,11),RF(1)

C   SET UP REDUCED FORM OF THE OPTIMIZATION MODEL FOR
C   REFERENCE STRATEGY EVALUATION

      NCON=NCON+2
      CALL CONT2(2,BIY,DBIY)
      DO 10 J=1,N
   10 CZ(J)=0.
      CZ(9)=-10.
      CZ(1)=-5.
      CZ(2)=-2.

C   SOLVE THE RESULTING L.P. PROBLEM USING THE TWO PHASE
C   SIMPLEX ALGORITHM

      CALL CANON
      CALL TPHSIM
      CALL LPSOL(0)
      DO 20 J=1,NF
   20 RF(J)=X(J)
      NCON=NCON-2
      RETURN
      END
```

2. TABLES

TABLE H - 1

EXPECTED COST ESTIMATION DATA

SENSITIVE PARAMETER SPECIFICATIONS

```
         6
         2        2         6        10         1         2        10         1
0.85          0.0004     0.95
         1        2         1         8         1         2         1         2
0.85          0.0004     1.0
         1        2         8         9         2         2         5         8
0.8           0.0004     0.9
         4        1         4        11         5         1         2         4
0.95          0.0001     1.0
         1        4         3         4         0
0.9           0.0004     1.0
         1        2         9         1         1         2         9        10
0.1           0.0004     1.0
```

M.C. SIMULATION CONTROL DATA

```
   250        25
  1.96         2.0    E-05   0.95
     0       907
```

TABLE H-2.

Printout of Initial Conditions

```
EVALUATION OF EXPECTED COST OF UNCERTAINTY
BY MONTE CARLO SIMULATION


THE SENSITIVE FACTORS ARE -

D( 2, 6,10)          MEAN =     0.8500          VARIANCE =     0.0004
     WITH DEPENDENT A( 2,10, 1)
A( 2, 1, 8)          MEAN =     0.8500          VARIANCE =     0.0004
     WITH DEPENDENT A( 2, 1, 2)
A( 2, 8, 9)          MEAN =     0.8000          VARIANCE =     0.0004
     WITH DEPENDENT D( 2, 5, 8)
A( 1, 4,11)          MEAN =     0.9500          VARIANCE =     0.0001
     WITH DEPENDENT D( 1, 2, 4)
A( 4, 3, 4)          MEAN =     0.9000          VARIANCE =     0.0004
A( 2, 9, 1)          MEAN =     0.1000          VARIANCE =     0.0004
     WITH DEPENDENT A( 2, 9,10)


MAXIMUM NO. OF SAMPLES THIS RUN =  250
ALPHA =     1.9600
DELTA =    0.2000E-04
CONFIDENCE LEVEL =     0.9500
```

TABLE H-3.

Examples of Detailed Sample Results

```
SAMPLE NO.     2

SENSITIVE FACTORS

D( 2, 6,10) =     0.8782
A( 2,10, 1) =     0.0718
A( 2, 1, 8) =     0.8710
A( 2, 1, 2) =     0.1290
A( 2, 8, 9) =     0.8236
D( 2, 5, 8) =     0.0764
A( 1, 4,11) =     0.9528
A( 2, 4,11) =     0.9528
A( 3, 4,11) =     0.9528
A( 4, 4,11) =     0.9528
D( 1, 2, 4) =     0.0472
D( 2, 2, 4) =     0.0472
D( 3, 2, 4) =     0.0472
D( 4, 2, 4) =     0.0472
A( 4, 3, 4) =     0.9014
A( 2, 9, 1) =     0.1048
A( 2, 9,10) =     0.8952

VALUE OF THE REFERENCE STRATEGY = -0.2841E-01

OPTIMAL STRATEGY FOR THIS STATE OF NATURE -

 OBJECTIVE FUNCTION = -0.2856E-01

 VARIABLE NAME              VALUE

          FX1             0.81434
          FX2             0.10500
          FX3             0.0
          FX4             0.0
          FX5             0.0
          FX6             0.03800
          FX7             0.0
          FX8             0.0
          FX9             0.01564
          FX10            0.0
          FX11            0.00429

THE REGRET =   0.1424E-03
```

TABLE H-3 Continued

SAMPLE NO. 6

SENSITIVE FACTORS

```
D( 2, 6,10) =    0.8760
A( 2,10, 1) =    0.0740
A( 2, 1, 8) =    0.8189
A( 2, 1, 2) =    0.1811
A( 2, 8, 9) =    0.8201
D( 2, 5, 8) =    0.0799
A( 1, 4,11) =    0.9428
A( 2, 4,11) =    0.9428
A( 3, 4,11) =    0.9428
A( 4, 4,11) =    0.9428
D( 1, 2, 4) =    0.0572
D( 2, 2, 4) =    0.0572
D( 3, 2, 4) =    0.0572
D( 4, 2, 4) =    0.0572
A( 4, 3, 4) =    0.9334
A( 2, 9, 1) =    0.0770
A( 2, 9,10) =    0.9230
```

VALUE OF THE REFERENCE STRATEGY = -0.2800E-01

OPTIMAL STRATEGY FOR THIS STATE OF NATURE -

OBJECTIVE FUNCTION = -0.2800E-01

VARIABLE NAME	VALUE
FX1	0.83997
FX2	0.08959
FX3	0.0
FX4	0.0
FX5	0.0
FX6	0.03800
FX7	0.0
FX8	0.0
FX9	0.01517
FX10	0.0
FX11	0.0

THE REGRET = 0.0

TABLE H-4.

Listing of Sample Points

NO. OF SAMPLE	REGRET	COST CERTAINTY	COST REF. STRATEGY
1	0.2647E-03	-0.2553E-01	-0.2526E-01
2	0.1424E-03	-0.2856E-01	-0.2841E-01
3	0.3247E-03	-0.2310E-01	-0.2278E-01
4	0.1118E-07	-0.2869E-01	-0.2869E-01
5	-0.3725E-08	-0.2664E-01	-0.2664E-01
6	0.0	-0.2800E-01	-0.2800E-01
7	0.3725E-08	-0.2779E-01	-0.2779E-01
8	-0.3725E-08	-0.2606E-01	-0.2606E-01
9	-0.1118E-07	-0.2644E-01	-0.2644E-01
10	-0.7451E-08	-0.2602E-01	-0.2602E-01
11	0.1073E-03	-0.2808E-01	-0.2797E-01
12	-0.3725E-08	-0.2672E-01	-0.2672E-01
13	-0.1863E-07	-0.2757E-01	-0.2757E-01
14	0.3901E-03	-0.2525E-01	-0.2486E-01
15	-0.3725E-07	-0.2671E-01	-0.2671E-01
16	-0.7451E-08	-0.2757E-01	-0.2757E-01
17	0.3725E-08	-0.2725E-01	-0.2725E-01
18	0.2059E-03	-0.2491E-01	-0.2471E-01
19	0.1622E-03	-0.2688E-01	-0.2672E-01
20	0.1397E-03	-0.2632E-01	-0.2619E-01
21	0.1091E-03	-0.2550E-01	-0.2539E-01
22	0.1561E-03	-0.2675E-01	-0.2659E-01
23	0.3246E-03	-0.2782E-01	-0.2750E-01
24	0.2193E-03	-0.2548E-01	-0.2526E-01
25	0.7766E-04	-0.2682E-01	-0.2675E-01
26	0.3151E-03	-0.2593E-01	-0.2562E-01
27	-0.1118E-07	-0.2860E-01	-0.2860E-01
28	0.0	-0.2802E-01	-0.2802E-01
29	0.2316E-03	-0.2821E-01	-0.2798E-01
30	0.2405E-03	-0.2579E-01	-0.2555E-01
31	0.7806E-04	-0.2542E-01	-0.2534E-01
32	-0.3725E-08	-0.2511E-01	-0.2511E-01
33	0.4142E-03	-0.2605E-01	-0.2564E-01
34	-0.2608E-07	-0.2680E-01	-0.2680E-01
35	0.2886E-03	-0.2749E-01	-0.2720E-01
36	0.1565E-06	-0.2747E-01	-0.2747E-01
37	0.1956E-03	-0.2727E-01	-0.2707E-01
38	0.7451E-08	-0.2718E-01	-0.2718E-01
39	0.3552E-03	-0.2776E-01	-0.2741E-01
40	0.9568E-04	-0.2669E-01	-0.2659E-01
41	-0.2980E-07	-0.2787E-01	-0.2787E-01
42	0.2288E-03	-0.2602E-01	-0.2579E-01
43	0.2689E-03	-0.2722E-01	-0.2695E-01
44	-0.7451E-08	-0.2766E-01	-0.2766E-01
45	0.3189E-03	-0.2426E-01	-0.2394E-01
46	0.3725E-08	-0.2749E-01	-0.2749E-01
47	0.2645E-03	-0.2381E-01	-0.2354E-01
48	0.0	-0.2647E-01	-0.2647E-01
49	0.3717E-03	-0.2255E-01	-0.2218E-01
50	0.7524E-04	-0.2602E-01	-0.2595E-01
51	0.2441E-03	-0.2610E-01	-0.2586E-01
52	0.2563E-04	-0.2698E-01	-0.2695E-01
53	0.0	-0.2769E-01	-0.2769E-01
54	-0.1490E-07	-0.2822E-01	-0.2822E-01

TABLE H-4 Continued

55	-0.3725E-08	-0.2638E-01	-0.2638E-01
56	-0.3353E-07	-0.2770E-01	-0.2770E-01
57	-0.3725E-08	-0.2688E-01	-0.2688E-01
58	-0.2608E-07	-0.2777E-01	-0.2777E-01
59	0.5015E-04	-0.2701E-01	-0.2696E-01
60	-0.7451E-08	-0.2673E-01	-0.2673E-01
61	0.1118E-07	-0.2736E-01	-0.2736E-01
62	0.1437E-03	-0.2748E-01	-0.2734E-01
63	0.2713E-03	-0.2681E-01	-0.2654E-01
64	0.1330E-03	-0.2643E-01	-0.2630E-01
65	0.9433E-04	-0.2726E-01	-0.2717E-01
66	0.3679E-03	-0.2662E-01	-0.2625E-01
67	0.1118E-07	-0.2879E-01	-0.2879E-01
68	-0.2980E-07	-0.2800E-01	-0.2800E-01
69	0.7451E-08	-0.2823E-01	-0.2823E-01
70	0.3725E-08	-0.2824E-01	-0.2824E-01
71	0.2648E-03	-0.2721E-01	-0.2694E-01
72	0.3725E-08	-0.2671E-01	-0.2671E-01
73	0.2425E-03	-0.2653E-01	-0.2629E-01
74	0.7451E-08	-0.2647E-01	-0.2647E-01
75	0.2410E-03	-0.2535E-01	-0.2511E-01
76	0.1475E-03	-0.2565E-01	-0.2550E-01
77	0.3679E-03	-0.2535E-01	-0.2499E-01
78	-0.3725E-08	-0.2696E-01	-0.2696E-01
79	0.3286E-03	-0.2702E-01	-0.2670E-01
80	-0.3725E-07	-0.2663E-01	-0.2663E-01
81	0.2578E-03	-0.2653E-01	-0.2627E-01
82	0.5461E-04	-0.2765E-01	-0.2760E-01
83	0.2743E-03	-0.2731E-01	-0.2704E-01
84	0.4052E-03	-0.2256E-01	-0.2216E-01
85	0.3725E-08	-0.2635E-01	-0.2635E-01
86	0.1118E-07	-0.2681E-01	-0.2681E-01
87	0.2595E-03	-0.2569E-01	-0.2543E-01
88	0.4201E-03	-0.2587E-01	-0.2545E-01
89	0.3568E-03	-0.2705E-01	-0.2669E-01
90	0.0	-0.2493E-01	-0.2493E-01
91	-0.1490E-07	-0.2856E-01	-0.2856E-01
92	0.2075E-04	-0.2533E-01	-0.2531E-01
93	0.3346E-03	-0.2584E-01	-0.2550E-01
94	0.1118E-07	-0.2697E-01	-0.2697E-01
95	0.3571E-03	-0.2641E-01	-0.2606E-01
96	0.0	-0.2700E-01	-0.2700E-01
97	0.3725E-08	-0.2571E-01	-0.2571E-01
98	0.1107E-03	-0.2614E-01	-0.2603E-01
99	0.3866E-04	-0.2565E-01	-0.2561E-01
100	0.3186E-03	-0.2710E-01	-0.2678E-01
101	-0.7451E-08	-0.2799E-01	-0.2799E-01
102	0.5032E-04	-0.2628E-01	-0.2623E-01
103	0.0	-0.2690E-01	-0.2690E-01
104	0.1319E-03	-0.2629E-01	-0.2616E-01
105	0.4466E-03	-0.2427E-01	-0.2382E-01
106	0.1078E-03	-0.2843E-01	-0.2832E-01
107	0.7451E-08	-0.2800E-01	-0.2800E-01
108	0.9311E-04	-0.2705E-01	-0.2696E-01
109	-0.1490E-07	-0.2630E-01	-0.2630E-01
110	0.8750E-04	-0.2748E-01	-0.2739E-01
111	-0.7451E-08	-0.2647E-01	-0.2647E-01
112	-0.3725E-08	-0.2513E-01	-0.2513E-01
113	0.4074E-03	-0.2512E-01	-0.2471E-01
114	-0.7451E-08	-0.2758E-01	-0.2758E-01

TABLE H-4 Continued

115	0.3858E-03	-0.2425E-01	-0.2387E-01
116	0.2321E-03	-0.2672E-01	-0.2649E-01
117	0.0	-0.2638E-01	-0.2638E-01
118	0.3418E-03	-0.2352E-01	-0.2318E-01
119	0.1632E-03	-0.2445E-01	-0.2429E-01
120	0.4000E-03	-0.2399E-01	-0.2359E-01
121	0.4818E-04	-0.2555E-01	-0.2550E-01
122	0.4605E-03	-0.2588E-01	-0.2542E-01
123	0.9909E-05	-0.2491E-01	-0.2490E-01
124	0.4147E-03	-0.2539E-01	-0.2498E-01
125	0.0	-0.2662E-01	-0.2662E-01
126	0.1339E-03	-0.2708E-01	-0.2695E-01
127	0.1505E-03	-0.2583E-01	-0.2568E-01
128	0.7451E-08	-0.2863E-01	-0.2863E-01
129	0.2968E-03	-0.2361E-01	-0.2331E-01
130	0.4312E-03	-0.2577E-01	-0.2534E-01
131	0.2591E-03	-0.2754E-01	-0.2728E-01
132	0.1763E-03	-0.2669E-01	-0.2651E-01
133	0.2755E-03	-0.2765E-01	-0.2738E-01
134	0.1007E-03	-0.2580E-01	-0.2570E-01
135	0.2140E-03	-0.2554E-01	-0.2533E-01
136	0.3705E-03	-0.2774E-01	-0.2737E-01
137	0.0	-0.2570E-01	-0.2570E-01
138	-0.3353E-07	-0.2747E-01	-0.2747E-01
139	0.7818E-04	-0.2700E-01	-0.2692E-01
140	0.1584E-03	-0.2564E-01	-0.2548E-01
141	0.6553E-05	-0.2549E-01	-0.2548E-01
142	0.1762E-03	-0.2606E-01	-0.2588E-01
143	-0.7451E-08	-0.2726E-01	-0.2726E-01
144	-0.3725E-08	-0.2532E-01	-0.2532E-01
145	-0.2980E-07	-0.2642E-01	-0.2642E-01
146	-0.7451E-08	-0.2675E-01	-0.2675E-01
147	0.3725E-08	-0.2858E-01	-0.2858E-01
148	-0.3725E-08	-0.2670E-01	-0.2670E-01
149	0.2533E-03	-0.2561E-01	-0.2536E-01
150	0.2185E-03	-0.2527E-01	-0.2505E-01
151	0.0	-0.2715E-01	-0.2715E-01
152	0.1970E-03	-0.2861E-01	-0.2842E-01
153	0.3725E-08	-0.2690E-01	-0.2690E-01
154	0.3725E-08	-0.2510E-01	-0.2510E-01
155	0.3356E-03	-0.2586E-01	-0.2553E-01
156	0.5953E-04	-0.2561E-01	-0.2555E-01
157	-0.1863E-07	-0.2725E-01	-0.2725E-01
158	0.2715E-03	-0.2664E-01	-0.2637E-01
159	0.2288E-03	-0.2491E-01	-0.2468E-01
160	0.7962E-04	-0.2572E-01	-0.2564E-01
161	-0.2980E-07	-0.2866E-01	-0.2866E-01
162	0.3276E-03	-0.2358E-01	-0.2326E-01
163	0.2844E-03	-0.2514E-01	-0.2486E-01
164	0.4258E-03	-0.2583E-01	-0.2540E-01
165	0.2528E-03	-0.2731E-01	-0.2706E-01
166	0.2261E-03	-0.2485E-01	-0.2462E-01
167	0.1808E-03	-0.2552E-01	-0.2534E-01
168	-0.7451E-08	-0.2581E-01	-0.2581E-01
169	0.4355E-05	-0.2759E-01	-0.2758E-01
170	0.2153E-03	-0.2712E-01	-0.2691E-01
171	0.1219E-03	-0.2590E-01	-0.2578E-01
172	-0.3725E-07	-0.2771E-01	-0.2771E-01
173	0.2364E-03	-0.2457E-01	-0.2434E-01
174	0.2800E-03	-0.2550E-01	-0.2522E-01

TABLE H-4 Continued

175	0.3288E-03	-0.2668E-01	-0.2635E-01
176	-0.3725E-08	-0.2483E-01	-0.2483E-01
177	0.1490E-07	-0.2726E-01	-0.2726E-01
178	0.1964E-03	-0.2594E-01	-0.2574E-01
179	0.3725E-08	-0.2673E-01	-0.2673E-01
180	0.4183E-03	-0.2469E-01	-0.2428E-01
181	0.7451E-08	-0.2837E-01	-0.2837E-01
182	0.6591E-04	-0.2597E-01	-0.2591E-01
183	0.3016E-03	-0.2467E-01	-0.2437E-01
184	-0.1118E-07	-0.2661E-01	-0.2661E-01
185	0.2027E-03	-0.2778E-01	-0.2757E-01
186	0.2387E-03	-0.2692E-01	-0.2668E-01
187	0.2325E-03	-0.2485E-01	-0.2462E-01
188	0.2526E-03	-0.2816E-01	-0.2790E-01
189	0.0	-0.2809E-01	-0.2809E-01
190	0.8819E-04	-0.2656E-01	-0.2648E-01
191	0.6198E-04	-0.2557E-01	-0.2551E-01
192	0.3260E-04	-0.2715E-01	-0.2712E-01
193	0.7451E-08	-0.2693E-01	-0.2693E-01
194	0.0	-0.2821E-01	-0.2821E-01
195	0.4832E-03	-0.2568E-01	-0.2520E-01
196	-0.7451E-08	-0.2710E-01	-0.2710E-01
197	0.1427E-04	-0.2626E-01	-0.2624E-01
198	0.0	-0.2706E-01	-0.2706E-01
199	0.2272E-03	-0.2556E-01	-0.2533E-01
200	0.2188E-03	-0.2409E-01	-0.2387E-01

3. DOCUMENTATION

3.1 ECOST

Subroutine ECOST performs a Monte Carlo simulation to estimate the expected cost of uncertainty in specified model parameters, controlling the generation of random samples, ensuring accuracy as outlined in Appendix A, calculating the required statistics, and printing a summary of the results. The variables in the COMMON block /EC/ are described below; those which must be defined prior to entry are marked with an asterisk

RM(I)*	expected value, I^{th} sensitive parameter
S(I)*	variance, I^{th} sensitive parameter
AT(I)*	upper limit - I^{th} sensitive parameter
SUM(J)	workspace
STAT(J)	storage of statistics
ALPHA*	α - defined Appendix A, section 1
DELTA*	required precision of estimate
CONF*	confidence level required
NTS(L,8)*	sensitive parameter definition, L^{th} parameter(NTS L,1) - type of base parameter .
	= 1 - a(I,J,K)
	= 2 - a(i,J,K)

= 3 - d(I,J,K)

= 4 - d(i,J,K)

= 5 - CON(I)

NTS (L,5) - type of coupled parameter as above

= 0 - no coupled parameter

NTS(L,2), NTS(L,6) - I

NTS(L,3), NTS(L,7) - J

NTS(L,4), NTS(L,8) - K

NSF* no. of sensitive parameters

MAXSAM* maximum no. of samples this run

NINSAM* minimum no. of samples before calculating any statistics

NSAM* no. of samples taken

IX odd integer for random number generator

NREJ no. of samples rejected.

3.2 EVAL, AVAL, INIT2, REFEED

These subroutines are adequately documented by comment cards included with the listings. Subroutine AVAL uses IBM's subroutine RM (32) for random number generation.

3.3 Mainline E.C.

This program estimates the expected cost of uncertainty using Monte Carlo simulation as presented in chapter III, section F, and Appendix A, section 1.

The input data requred are: model data similar to that listed in table C-1, defined in table C-3, but with optimal split factors; variable names for LPSOL, listed in table D-1; and expected cost estimation data as listed in table H-1. The mainline produces a summary of initial conditions, which appears as table H-2. Examples of the detailed sample results produced by EVAL appear as table H-3 and a summary listing of sample results produced by ECOST appears as table H-4. The summary of expected cost estimation results appears as table 18.

Subroutines required are documented in this appendix or as follows:

INPUT	-	Appendix C
INIT	-	Appendix E